



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**  
FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV AUTOMATIZACE A INFORMATIKY**  
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**KONSTRUKCE MOBILNÍHO ROBOTA PRO  
MONITOROVÁNÍ TEPLOTY OKOLÍ**  
THE CONSTRUCTION OF A MOBILE ROBOT FOR MONITORING AMBIENT  
TEMPERATURES

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. Štěpán Čejka**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. Daniel Zuth, Ph.D.**

**BRNO 2016**



## Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Bc. Štěpán Čejka</b>
Studijní program:	Strojní inženýrství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	<b>Ing. Daniel Zuth, Ph.D.</b>
Akademický rok:	2015/16

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

### **Konstrukce mobilního robota pro monitorování teploty okolí**

#### **Stručná charakteristika problematiky úkolu:**

Práce se zabývá konstrukcí autonomního mobilního robota, který je schopen detekovat zvýšenou teplotu v okolí, která může znamenat zvýšené nebezpečí požáru. Řídící funkce robota bude zajišťovat vývojový kit fy Freescale. Práce je součástí projektu Aplikovaná informatika a řízení FSI-S-14-2533

#### **Cíle diplomové práce:**

- 1) Popis prvků použitelných pro realizaci
- 2) Výběr vhodných prvků pro konstrukci robota
- 3) Konstrukce mobilního robota
- 4) Návrh vhodného algoritmu pro monitorování určitého objektu

#### **Seznam literatury:**

Matoušek, D. (2006): Práce s mikrokontroléry. BEN. Praha  
Novák, P. (2005): Mobilní roboty - pohony, senzory, řízení. BEN, Praha.  
Freescale Semiconductor, Freescale Semiconductor Inc, <http://www.freescale.com>, přístup 20. listopadu 2015.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2015/16

V Brně, dne

L. S

---

Ing. Jan Roupec, Ph.D.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Tato diplomová práce se zabývá návrhem a řízením mobilního pásového robota na základě informací získaných prostřednictvím infračerveného termočlánku. Součástí práce je implementace řídicího software firmwaru pro řízení podvozku a komunikaci s použitými senzory. Funkcionalita navrženého systému je demonstrována na robotické úloze, kdy se robot aktivně snaží ve svém okolí lokalizovat zdroj nadměrného tepelného záření. Teoretická část práce se zabývá základním popisem běžných robotických podvozků, problematikou bezdotykového měření teploty, dále pak rozбором komponent použitých při stavbě robota a jejich principy. Praktická část práce se věnuje implementaci řídicího firmwaru a detailnímu popisu řešení vybraných problémů. V samém závěru práce jsou shrnuty dosažené výsledky a úspěšnost robota při detekci podezřelých objektů s vysokou povrchovou teplotou.

## **ABSTRACT**

This diploma thesis deals with the design and control of the mobile robot with caterpillar tracks on the basis of information obtained via infrared thermocouple. The work includes firmware implementation for robot chassis control and communication with the sensors used. The functionality of the proposed system is demonstrated on a robotic task when the robot is searching the source of excessive heat within his surroundings. The theoretical part deals with the basic description of the common robotic chassis, contactless temperature measurement, further analysis of the components used and their principles. The practical part is devoted to the firmware implementation and detailed description of selected problems. In the end of the thesis there is a summarization of the achievements and the success of the robot while detection of the suspicious objects with high surface temperature.

## **KLÍČOVÁ SLOVA**

Robot, Zumo, řízení, mikrokontrolér, Freescale, FRDM-KL25Z, software, motor, enkodér, PID regulace, infratermočlánek.

## **KEYWORDS**

Robot, Zumo, control, microcontroller, Freescale, FRDM-KL25Z, software, motor, encoder, PID control, infrared thermocouple.



## **BIBLIOGRAFICKÁ CITACE**

ČEJKA, Š. *Konstrukce mobilního robota pro monitorování teploty okolí*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 93 s. Vedoucí diplomové práce Ing. Daniel Zuth, Ph.D.





## **PODĚKOVÁNÍ**

Na tomto místě bych chtěl poděkovat Ing. Danielu Zuthovi, Ph.D. za vedení, cenné rady a připomínky během zpracování této závěrečné práce.



## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Daniela Zutha, Ph.D. a s použitím literatury uvedené v seznamu.

V Brně dne 27.5.2016

.....  
Štěpán Čejka



# OBSAH

<b>1</b>	<b>ÚVOD .....</b>	<b>17</b>
<b>2</b>	<b>KLASIFIKACE MOBILNÍCH ROBOTŮ .....</b>	<b>19</b>
2.1	Podvozky kolových a pásových mobilních robotů .....	19
2.1.1	Diferenciální podvozek .....	19
2.1.2	Ackermanův podvozek.....	20
2.1.3	Podvozky se všesměrovými koly .....	20
2.1.4	Pásové podvozky.....	20
<b>3</b>	<b>TEORIE BEZKONTAKTNÍHO SNÍMÁNÍ TEPLoty .....</b>	<b>21</b>
3.1	Infračervené záření .....	21
3.2	Teorie absolutně černého tělesa .....	21
3.2.1	Kirchhoffův zákon.....	22
3.2.2	Stefanův-Boltzmannův zákon .....	22
3.2.3	Wienův posunovací zákon .....	22
3.2.4	Planckův vyzařovací zákon .....	23
3.3	Radiace reálných povrchů .....	24
3.4	Detektory v pyrometrii .....	24
3.4.1	Termální detektory .....	24
3.4.2	Pyroelektrické detektory .....	24
3.4.3	Fotonové detektory.....	25
3.5	Vybrané typy pyrometrů .....	25
3.5.1	Radiační (úhrnné) pyrometry .....	25
3.5.2	Jasové (spektrální) pyrometry .....	25
3.5.3	Infračervené termočlánky .....	26
3.5.4	Vyhodnocení radiační teploty .....	26
<b>4</b>	<b>ŘÍDÍCÍ JEDNOTKA .....</b>	<b>27</b>
4.1	Mikrokontrolér .....	27
4.1.1	Základní parametry MCU .....	27
<b>5</b>	<b>POHONNÝ SUBSYSTÉM ROBOTA.....</b>	<b>29</b>
5.1	DC motory.....	29
5.1.1	Princip .....	29
5.1.2	Řízení otáček .....	30
5.1.3	Řízení smyslu otáček a vhodné budicí obvody .....	30
5.2	Krokové motory .....	32
5.2.1	Rozdělení krokových motorů podle konstrukce rotoru.....	32
5.2.2	Princip .....	33
5.2.3	Řízení krokových motorů.....	33
5.3	Servomotory .....	35
5.3.1	Řízení servomotoru .....	35
<b>6</b>	<b>SENZORICKÝ SUBSYSTÉM ROBOTA .....</b>	<b>37</b>
6.1	Rotační enkodéry .....	37
6.1.1	Fyzikální principy a detekční schopnosti rotačních enkodérů .....	37
6.1.2	Kvadrurní dekódování .....	39
6.2	MEMS gyroskopy .....	40
6.2.1	Vyhodnocení úhlu natočení.....	40
6.3	MEMS akcelerometry .....	41

6.3.1	Vyhodnocení úhlu náklonu .....	41
6.4	Taktilní senzory.....	41
6.5	Sonary .....	42
6.5.1	Stanovení vzdálenosti překážky od sonaru .....	43
<b>7</b>	<b>KOMUNIKAČNÍ SUBSYSTÉM ROBOTA.....</b>	<b>45</b>
7.1	Bluetooth .....	45
7.2	ZigBee .....	45
<b>8</b>	<b>NÁVRH ROBOTA .....</b>	<b>47</b>
8.1	Podvozek .....	47
8.1.1	Pásový podvozek Zumo Robot .....	47
8.1.2	Parametry podvozku Zumo Robot v1.2 .....	48
8.2	Mikro DC motory Pololu .....	49
8.3	Magnetický enkodér Pololu .....	50
8.4	Řídící jednotka .....	51
8.4.1	Vývojový kit FRDM-KL25Z .....	51
8.4.2	Mikrokontrolér MKL25Z128VLK4.....	52
8.5	Sonar HC-SR04.....	52
8.5.1	Obsluha sonaru .....	53
8.5.2	Převodník úrovní .....	53
8.6	Gyroskop STMicroelectronics L3GD20H .....	53
8.7	TPA81 Thermopile Array .....	55
8.7.1	Komunikace a obsluha .....	55
8.7.2	Vyhodnocení radiální teploty .....	56
8.8	Bluetooth modul HC-05 .....	57
8.9	Rozšiřující deska pro Zumo Shield a FRDM-KL25Z.....	58
<b>9</b>	<b>REALIZACE ŘÍDÍCÍHO FIRMWARE .....</b>	<b>59</b>
9.1	Task scheduler.....	59
9.2	Řízení robota .....	60
9.2.1	Rozjezd robota.....	61
9.2.2	Řízení jízdy – regulace rychlosti a polohy .....	62
9.2.3	Otáčení robota .....	66
9.3	Měření rychlosti .....	66
9.4	Měření natočení a obsluha gyroskopu.....	69
9.4.1	Inicializace gyroskopu.....	69
9.4.2	Kalibrace gyroskopu .....	69
9.4.3	Výpočet úhlu natočení.....	70
9.5	Měření okolní teploty .....	70
9.6	Měření povrchové teploty, detekce podezřelého objektu a překážky .....	71
9.6.1	Krokovací režim serva .....	71
9.6.2	Detekční algoritmus .....	72
9.6.3	Regulace polohy serva .....	74
9.6.4	Korekce směru jízdy v okamžiku detekce.....	75
9.6.5	Korekce směru jízdy ve sledovacím režimu.....	76
9.6.6	Měření vzdálenosti od překážky .....	77
9.7	Komunikace .....	78
<b>10</b>	<b>ZHODNOCENÍ A DISKUZE .....</b>	<b>79</b>
<b>11</b>	<b>ZÁVĚR.....</b>	<b>81</b>
<b>12</b>	<b>SEZNAM POUŽITÝCH ZDROJŮ .....</b>	<b>83</b>

<b>13</b>	<b>SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK.....</b>	<b>87</b>
13.1	Seznam symbolů a zkratk.....	87
13.2	Seznam obrázků .....	88
13.3	Seznam tabulek .....	90
<b>14</b>	<b>SEZNAM PŘÍLOH .....</b>	<b>91</b>





# 1 ÚVOD

Lidé se od nepaměti zabývali otázkou, jak si co nejvíce usnadnit nebo se úplně oprostít od fyzicky namáhavé, nebezpečné či stereotypní práce. Pomineme-li delegaci práce na jinou osobu, tak bylo možné problém s ulehčováním automatizovatelné práce řešit využitím důmyslných technických zařízení. Z počátku se jednalo pouze o mechanické stroje, později byly tyto stroje doplněny o elektronické prvky a aktuálního vrcholu bylo dosaženo nasazením počítačů do role řídicí jednotky celého zařízení.

Obor, který se zabývá vývojem těchto zařízení, se nazývá robotika. Robotika je kombinace několika technických disciplín. Především se zde uplatňuje mechanika, elektrotechnika, senzorika, automatizace, informatika a umělá inteligence. Problematikou robotiky se zabývá i tato práce, detailněji se pak práce zaměřuje především na oblast automatizace a řízení autonomního mobilního robota.

Mým úkolem je navrhnout a vyvinout autonomní robot pro monitorování a vyhledávání objektů s potenciálně nebezpečně vysokou povrchovou teplotou. Návrh robota spočívá ve výběru podvozku, řídicí jednotky, motorů pro pohon robota a sady senzorů pro bezkontaktní měření radiační teploty a sledování pohybu robota v prostoru. Součástí práce je vývoj SW pro řízení podvozku na základě dat získaných z použitých senzorů.

Navržený robot je založen na pásovém podvozku Zumo. Podvozek je doplněn senzory pro měření rychlosti a směru pohybu. Obě veličiny jsou řízeny PID regulátory. Dále je podvozek vybaven ultrazvukovým sonarem pro detekci překážek a radiačním pyrometrem typu infračervený termočlánek pro měření radiační teploty. Při návrhu a vývoji robota bylo přihlíženo k tomu, aby celá konstrukce byla snadno replikovatelná a použitelná jako školní výukový model.

V rešeršní části práce jsou s využitím odborné literatury popsány principy bezdotykového měření povrchové teploty a dále pak popis vybraných typů podvozků, pohonů a senzorů, které se běžně využívají pro realizaci robotů.

V dalších kapitolách jsou pak představeny zvolené komponenty včetně jejich parametrů, popis řídicích algoritmů, rozbor problémů řešených při vývoji a způsob jejich řešení.



## 2 KLASIFIKACE MOBILNÍCH ROBOTŮ

Mobilní roboty je možné rozlišovat dle několika kritérií. Z pohledu řízení je možné rozdělení na dva základní typy [1]:

- **Autonomní roboty** – jsou schopni samostatně vykonávat zadanou úlohu, pohybovat se a orientovat se v neznámém prostředí a vyhnout se nebezpečným situacím bez zásahu člověka.
- **Dálkově řízené roboty** – jsou řízené operátorem, který má vizuální přehled o situaci v okolí robota. Za dálkové řízení se také považuje teleprezenční řízení s využitím prvků virtuální reality, kdy se operátor cítí jako by se přímo nacházel v prostředí, v němž robot pracuje.

Podle prostředí, ve kterém se robot pohybuje lze mobilní roboty klasifikovat na pozemní (vnitřní/vnější prostředí), vodní, létající a speciální roboty pracující v extrémním prostředí jako je např. vesmír. Tyto skupiny lze dále dělit dle způsobu pohybu. V případě pozemních robotů se jedná pak především o roboty kolové a pásové, plazící se či kráčející.

### 2.1 Podvozky kolových a pásových mobilních robotů

V oblasti mobilních robotů tvoří nejrozšířenější množinu podvozků podvozky kolové a pásové. Vzhledem k charakteru robotické úlohy, kterou má navrhovaný robot vykonávat se jeví tyto podvozky jako optimální a budou v následujícím textu rozebrány podrobněji.

Na kolovém podvozku rozlišujeme kola hnaná a vlečná. Hnaná slouží pro pohon podvozku, vlečná pak pro stabilizaci či změnu směru jízdy. Dalším důležitým parametrem kol je počet stupňů volnosti jejich pohybu. Kola s jedním stupněm volnosti se mohou pouze odvalovat. V případě kola s dvěma stupni volnosti se kolo může pohybovat podle dvou os – například přední kolo osobního auta, nebo kulička v případě že jsou obě osy rovnoběžné s povrchem [1].

#### 2.1.1 Diferenciální podvozek

Diferenciální podvozek je vybaven dvojicí nezávisle hnaných kol. Stabilita podvozku je pak zajišťována dalším kolem, kuličkou či třecími ploškami. Byli však vyvinuti i roboti bez těchto stabilizačních prvků, kteří dokáží balancovat ve vzpřímené poloze. Dále jsou známí např. čtyřkoloví roboti s nezávislým pohonem každého kola.

Tento typ podvozku vyniká manévrovatelností – robot je schopen otáčet se na místě kolem svislé osy. Řízení směru jízdy se provádí změnou rychlosti a smyslu otáčení hnaných kol [1].



Obr. 1) Balancující robot [1]

### 2.1.2 Ackermanův podvozek

Podvozek typicky obsahuje tři, čtyři i více kol, přičemž jedním či více koly lze natáčet a měnit tak směr jízdy a další sada kol slouží pro pohon. Je však možné zatáčet i hnanými koly jako u osobního automobilu. Tento podvozek se používá převážně u větších robotů. Nevýhodou je nemožnost otáčet se na místě. Robot se může otáčet pouze po trajektorii s určitým minimálním poloměrem [1].

### 2.1.3 Podvozky se všesměrovými koly

Tyto podvozky dovolují pohyb ve všech směrech. Změnu směru umožňují okamžitě s nulovým poloměrem otáčení. Nevýhodou jsou vysoké náklady na výrobu precizních všesměrových kol a podvozky nejsou vhodné pro zdolávání překážek. Lze se s nimi setkat v experimentálních aplikacích, v profesionálních aplikacích se pak uplatňují jako různé montážní plošiny a manipulátory [1].



Obr. 2) Detail všesměrového kola [27]

### 2.1.4 Pásové podvozky

Jedná se v podstatě o variantu diferenciálních podvozků s tím, že kola slouží pro vedení pásů. Díky pásům podvozek vykazuje výbornou terénní průchodnost a stoupavost a umožňuje otáčení na místě. Nevýhodou je, že každá změna směru je doprovázena prokluzem pásů. Ve srovnání s kolovými podvozky je pohon pásových podvozků energeticky náročnější a nelze použít metodu odometrie pro určování polohy robota. Tento podvozek nachází uplatnění především v terénu.

Často bývá používán ve vojenských aplikacích nebo v průzkumných či hasičských robotech. Ve vnitřním prostředí ho lze s výhodou použít pro zdolávání schodů a jiných obtížných překážek. Často je robot na pásovém podvozku dálkově ovládán operátorem [1].



Obr. 3) Vojenský robot TALON s pásovým podvozkiem [28]

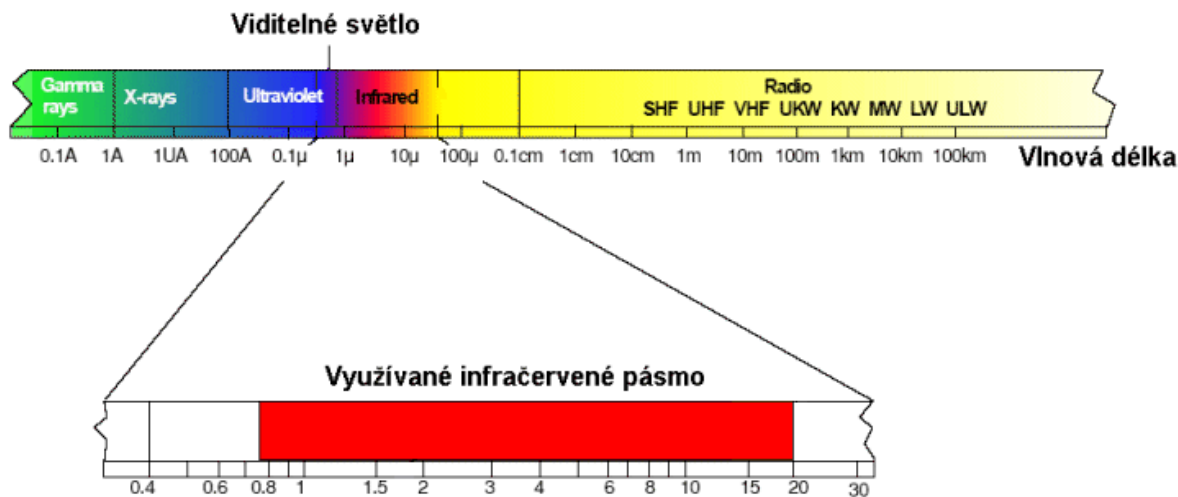
### 3 TEORIE BEZKONTAKTNÍHO SNÍMÁNÍ TEPLOTY

K tomu, aby robot mohl monitorovat své okolí a vyhledávat objekty s potenciálně nebezpečnou teplotou, musel být vybaven vhodným snímačem. Klíčové pro tuto úlohu bylo použití bezkontaktního snímání teploty. Po tomto účel jsem se rozhodl použít infračervený termočlánek. Považuji za vhodné na úvod uvést fyzikální zákonitosti využívané při bezkontaktním měření teploty a představit některé typy bezdotykových detektorů teploty.

#### 3.1 Infračervené záření

Všechny látky a tělesa vyzařují do okolí energii ve formě elektromagnetického záření. Intenzita tohoto záření se mění s teplotou tělesa a do určité teploty není toto záření pro lidské oko viditelné, neboť leží v tzv. infračervené oblasti. Princip bezkontaktních snímačů teploty spočívá právě v detekci elektromagnetického záření, přičemž některé vyhodnocují pouze jeho infračervenou oblast. Za objevem infračerveného záření stojí anglický vědec a astronom Frederick William Herschel [3].

Při experimentu, kdy pozoroval Slunce přes barevná skla, si všiml, že tepelný vjem není u všech skel stejný. V důsledku toho tedy Herschel provedl experiment, kdy pomocí optických hranolů rozložil světelné záření na jednotlivé složky a tyto pak testoval konvenčním rtuťovým teploměrem. Zjistil, že čím více teploměr posouval k červené oblasti spektra, tím byla teplota vyšší. Po posunutí teploměru až za červenou oblast očekával, že tepelný efekt zmizí. Místo toho však teploměr naměřil vyšší teploty, než u kterékoliv jiné viditelné složky. Tato oblast teda byla pojmenována infračervená (tj. „pod červená“) [4].



Obr. 4) Infračervená oblast elektromagnetického záření [5]

#### 3.2 Teorie absolutně černého tělesa

Kromě emitace záření každé těleso také záření pohlcuje nebo ho odráží. Množství pohlceného záření reálným tělesem závisí na jeho barvě a materiálu, přičemž platí, že černá a matná tělesa pohlcují záření nejlépe. Pro snadnější popis procesů pohlcování a vyzařování byla zavedena G. R. Kirchhoffem fyzikální abstrakce, tzv. absolutně černé těleso.

### 3.2.1 Kirchhoffův zákon

V roce 1859 Gustav Robert Kirchhoff a Robert Bunsen zavedli teorii obecné absorpce a radiace známou jako Kirchhoffův-Bunsenův zákon. Tento zákon říká, že schopnost látky emitovat záření závisí pouze na teplotě a na schopnosti záření absorbovat, nikoliv na materiálu tělesa.

$$\frac{M_0}{\alpha} = f(T) \quad (1)$$

Kde  $M_0$  vyjadřuje schopnost vyzařování (tzv. emisivitu),  $\alpha$  je schopnost pohlcovat záření (tzv. absorpce) a  $f(T)$  je funkce termodynamické teploty.

Jako důsledek tohoto zákona definoval Kirchhoff pojem absolutně černého tělesa. Jedná se o fyzikální model, který dokonale pohlcuje veškeré dopadající elektromagnetické záření a žádné neodráží ani nepropouští. Těleso si lze představit jako dutinu s černým matným povrchem, do které malou štěrbinou vstupuje záření. Toto záření je uvnitř tělesa zcela pohlceno. Čím více elektromagnetického záření je tímto tělesem pohlceno, tím více vzroste jeho teplota a tím více záření bude emitovat. Pro toto ideální těleso pak lze vztah obecného Kirchhoffova zákona zjednodušit:

$$M_0 = f(T) \quad (2)$$

Schopnost vyzařování tohoto ideálního tělesa pak závisí čistě na jeho teplotě. Problematikou nalezení neznámé funkce  $f(T)$  se zabývají další zákony níže.

### 3.2.2 Stefanův-Boltzmannův zákon

Tento zákon publikovaný roku 1879 Josefem Stefanem a Ludwigem Boltzmannem říká, že celková intenzita vyzařování (emisivita) černým tělesem roste se čtvrtou mocninou jeho absolutní termodynamické teploty.

$$M_0 = \sigma \cdot T^4 \quad (3)$$

Kde  $M_0$  je celková intenzita vyzařování absolutně černého tělesa,  $\sigma$  je tzv. Stefan-Boltzmannova konstanta ( $\sigma = 5,67 \cdot 10^{-8} \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-4}$ ) a  $T$  je termodynamická teplota [K].

Pro skutečná tzv. šedá tělesa mající reálné vlastnosti platí následující modifikace tohoto zákona:

$$M = \varepsilon \cdot M_0 \quad (4)$$

Kde  $\varepsilon$  je emisní součinitel viz kapitola 3.3.

### 3.2.3 Wienův posunovací zákon

Německý fyzik Wilhelm Wien v roce 1893 provedl sérii pokusů, při kterých sledoval distribuci vlnových délek záření absolutně černého tělesa. Výsledky vynesl do grafů popisující závislost intenzity záření na vlnové délce při různých teplotách. Všiml si, že špičková hodnota intenzity záření dané vlnové délky je přímo úměrná energii a nepřímo úměrná termodynamické teplotě.

Jinými slovy, s rostoucí teplotou se zvyšuje celková energie vyzařovaného záření (podle Stefanova-Boltzmannova zákona) a maximální hodnota spektrální hustoty tohoto záření se posouvá ke kratším vlnovým délkám. Důsledkem je pak změna barvy světelného záření z červené, přes žlutou až k bílé barvě.

$$\lambda_{max} = \frac{b}{T} \quad (5)$$

Kde  $\lambda_{max}$  je vlnová délka [m], u které je dosaženo nejvyšší intenzity vyzařování při dané teplotě,  $T$  je termodynamická teplota a  $b = 2,9 \cdot 10^{-3} \text{ m} \cdot \text{K}$  je konstanta.

### 3.2.4 Planckův vyzařovací zákon

Ani jeden z předchozích zákonů však nedefinuje funkci  $M_0(\lambda, T)$ , která popisuje intenzitu záření absolutně černého tělesa ve všech oblastech spektra elektromagnetického záření. Toto se povedlo až Maxi Planckovi. Planck přirovnal těleso k sadě harmonických oscilátorů a vyslovil hypotézu, že černé těleso nemůže záření pohlcovat a vyzařovat spojitě, ale pouze po diskrétních kvantech. Podle Plancka pak má každé kvantum určitou energii, která je přímo úměrná frekvenci záření.

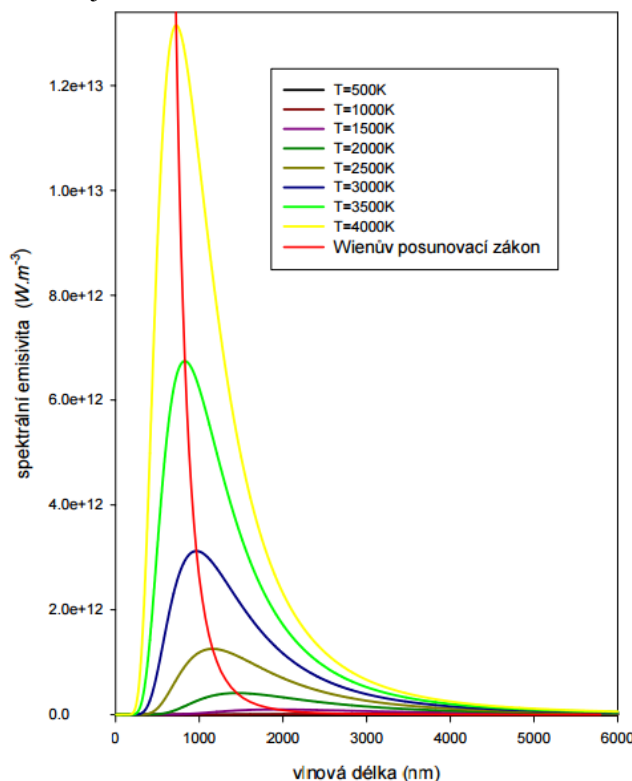
$$E = h \cdot \nu = \frac{h \cdot c}{\lambda} \quad (6)$$

Kde  $h = 6,626 \cdot 10^{-34} \text{ J} \cdot \text{s}$  je tzv. Planckova konstanta,  $\nu$  je frekvence záření [Hz],  $c$  je rychlost světla ve vakuu [ $\text{m} \cdot \text{s}^{-1}$ ] a  $\lambda$  je vlnová délka záření.

Na základě této hypotézy poté Planck odvodil zákon spektrálního rozdělení vyzařování černého tělesa:

$$M_0(\lambda, T) = \frac{2 \cdot \pi \cdot h \cdot c^2}{\lambda^5 \cdot (e^{\frac{h \cdot c}{k \cdot h \cdot T}} - 1)} \quad (7)$$

Kde  $h = 6,626 \cdot 10^{-34} \text{ J} \cdot \text{s}$  je Planckova konstanta,  $k = 1,380648 \cdot 10^{-23} \text{ J} \cdot \text{kg}^{-1}$  je Boltzmannova konstanta,  $c$  je rychlost světla ve vakuu a  $\lambda$  je vlnová délka záření.



Obr. 5) Planckův vyzařovací zákon [6]

### 3.3 Radiace reálných povrchů

Ideálních vlastností absolutně černého tělesa nelze v normálních podmínkách dosáhnout. Těleso z reálného materiálu část elektromagnetického záření absorbuje, část může propustit a část odrazí. S tím je spjata definice pojmů:

- **Absorptivita** – je označovaná jako  $\alpha$  a vyjadřuje poměr absorbovaného a celkového záření dopadajícího na těleso.
- **Reflexivita** – je označovaná jako  $\rho$  a vyjadřuje poměr odraženého vůči celkovému záření.
- **Transmisivita** – je označovaná jako  $\tau$  a vyjadřuje poměr propuštěného záření tělesem vůči celkovému záření.

Celkový součet všech dílčích energií, tj. absorbované, odražené a procházející musí být roven celkovému množství dopadajícího záření:

$$\alpha + \rho + \tau = 1 \quad (8)$$

Z pohledu velmi přesného bezkontaktního snímání teploty je však nejdůležitější schopnost tělesa tepelné záření vyzařovat. S touto schopností jsou spjaty pojmy:

- **Emisivita** – je označována jako  $\varepsilon$  a popisuje schopnost materiálu emitovat tepelné záření. Kirchhoff definoval emisivitu jako bezrozměrnou veličinu, která vyjadřuje poměr intenzity vyzařování reálného tělesa vůči intenzitě vyzařování absolutně černého tělesa. Emisivita je pouze jedním z faktorů emitance.
- **Emitance** – též označovaná jako emitivita, popisuje schopnost daného objektu emitovat tepelné záření. Faktory emitance jsou kromě emisivity také např. teplota, tvar či povrchová úprava objektu.

Pro absolutně černé těleso platí  $\alpha = \varepsilon = 1$ ,  $\rho = \tau = 0$ . Čerpáno z [4].

### 3.4 Detektory v pyrometrii

Jako detektor je označována citlivá část snímače, která reaguje na měřenou veličinu a převádí ji na jinou, lépe vyhodnotitelnou veličinu. V pyrometrii, tedy pro bezkontaktní měření teploty, se uplatňují především detektory termální, fotonové a pyroelektrické. Čerpáno z [4].

#### 3.4.1 Termální detektory

Termální detektor absorbuje dopadající záření a dle jeho intenzity dochází k ohřívání detektoru. Detektory mají černou matnou barvu a reagují tak na záření v širokém spektru, avšak s menší citlivostí oproti fotonovým detektorům. Měření je relativně pomalé (*ms* až *s*), protože je nutné dosáhnout termodynamické rovnováhy při každé změně teploty měřeného objektu. Typickými zástupci jsou bolometry, infračervené termočlánky a termistory.

#### 3.4.2 Pyroelektrické detektory

Při konstrukci těchto detektorů jsou využívány speciální pyroelektrické materiály, které se při změnách teploty deformují. Deformace, která je úměrná dopadajícímu záření, tak vyvolá změnu povrchové náboje detektoru. Záření je na detektor soustředováno optickou soustavou. Pokud dojde k tepelné změně v zorném poli snímače, je na odpovídající části detektoru možné



detekovat změnu náboje. Tyto detektory se využívají především v pohybových čidlech, než pro přesné měření teploty.

### 3.4.3 Fotonové detektory

Tyto detektory v reakci na dopadající záření uvolňují elektrický náboj. Tento jev je opět měřen jako změna některé elektrické veličiny. Dle materiálu a provedení detektoru jde pak o změnu odporu nebo napětí na výstupu. Fotonové detektory vykazují mezní vlnovou délku, po jejímž překročení již neposkytují odezvu. Možnost měření pouze v omezeném pásmu je ale vyvážena vysokou citlivostí, která je až 100000 krát vyšší než u termálních detektorů. Doba odezvy se pohybuje v řádech  $\mu s$ . Nevýhodou pyrometrů s fotonovým detektorem je, že často potřebují chlazení. Mezi fotonové detektory patří fotorezistor, fotodioda a fototranzistor.

## 3.5 Vybrané typy pyrometrů

### 3.5.1 Radiační (úhrnné) pyrometry

Jsou to měřidla pracující v širokém pásmu vlnových délek elektromagnetického záření. Ve speciálních případech se lze setkat s měřicím rozsahem  $-100^{\circ}C$  až  $5000^{\circ}C$ . Záření emitované tělesem je optickou soustavou soustředováno na termočlánek či bolometr. Důležité je, aby plocha měřeného objektu překryla celé zorné pole snímače, jinak dochází k velkým chybám měření.

Teplota objektu je vyhodnocována přímo podle Stefan-Bolzmanova zákona. K převodu mezi změřenou a skutečnou teplotou objektu pak slouží vztah beroucí v potaz emisní součinitel povrchu objektu [33]:

$$T_S = \frac{T_P}{\sqrt[4]{\varepsilon}} \quad (9)$$

Kde  $T_S$  je skutečná teplota  $[K]$ ,  $T_P$  je teplota naměřená pyrometrem  $[K]$ ,  $\varepsilon$  je emisní součinitel.

### 3.5.2 Jasové (spektrální) pyrometry

Jasové pyrometry pracují na jedné vlnové délce, typicky  $\lambda = 0,65 \mu m$ . Ta odpovídá červené barvě světla. Měření probíhá tak, že se jas měřeného objektu porovnává s jasnem porovnávacího zdroje (žárovky) pyrometru. Jas porovnávací žárovky se nastavuje tak dlouho, až se vlákno žárovky ztratí v záření vyzařovaném měřeným objektem. Porovnávání se provádí buď manuálně lidským okem, nebo automaticky. Rozsah takového pyrometru činí cca  $700^{\circ}C$  až  $1500^{\circ}C$ . S použitím šedého filtru před objektivem lze zvýšit rozsah na cca  $1300^{\circ}C$  až  $3500^{\circ}C$ . Jasový pyrometr nachází uplatnění například ve slévárnách.

Teplota odečtená na stupnici pyrometru odpovídá záření černého tělesa a bývá obvykle nižší, než skutečná teplota. Je třeba ji tedy korigovat a pro skutečnou teplotu platí následující vztah [33]:

$$\frac{1}{T_S} = \frac{1}{T_P} + \frac{\lambda}{C_2} \cdot \ln \varepsilon \quad (10)$$

Kde  $T_S$  je skutečná teplota,  $T_P$  je teplota naměřená pyrometrem,  $\lambda$  použitá vlnová délka,  $C_2 = 14,388 \cdot 10^{-3} m \cdot K$  je druhá vyzařovací konstanta Planckova zákona a  $\varepsilon$  je emisní součinitel.

### 3.5.3 Infračervené termočlánky

Tato skupina pyrometrů pracuje obdobným způsobem jako standartní termočlánky. Termoelektrický efekt, objevený T. J. Seebeckem, spočívá ve spojení dvou vodičů z různých kovů (např. železo a konstantan) do uzavřeného elektrického obvodu. Jsou-li oba konce (měřicí a srovnávací) vystaveny rozdílné teplotě, tak na srovnávacím spoji lze naměřit elektromotorické napětí v řádech mV.

Infračervené termočlánky nevyžadují napájení. Ke generování signálu nesoucího informaci o teplotě využívají pouze dopadající infračervenou radiaci a mechanismus termoelektrického jevu. Teplotní rozsah činí cca  $-45\text{ }^{\circ}\text{C}$  až  $2800\text{ }^{\circ}\text{C}$  s přesností až  $0,01\text{ }^{\circ}\text{C}$ .

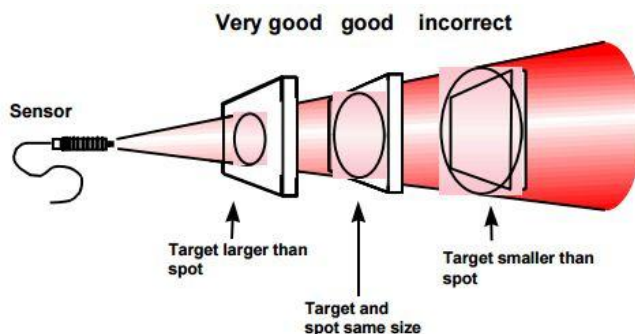
Před měřením by měl být infračervený termočlánek, stejně jako ostatní měřidla založené na vyhodnocování radiace, správně kalibrován podle specifických vlastností povrchu měřeného objektu. Kalibrace se provádí změřením teploty objektu nezávislým teploměrem a následně se nastavuje emisní součinitel termočlátku, dokud měřená teplota neodpovídá hodnotám nezávislého teploměru.



Obr. 6) Infračervený termočlánek firmy Omega [4]

### 3.5.4 Vyhodnocení radiační teploty

Při bezdotykovém měření teploty radiačními pyrometry či infračervenými termočlánky, je třeba pamatovat na to, že optický systém pyrometru zachycuje tepelné záření z určité měřené plochy, obecně libovolného tvaru. Přitom platí, že čím víc je teploměr od měřeného objektu vzdálen, tím je tato měřená plocha větší a snímač reaguje na veškeré záření tepelné záření z této oblasti. V případě, že je viditelná plocha měřeného objektu menší než zorné pole snímače, dochází k hrubým chybám při stanovení radiační teploty. Pro přesné měření tedy musí být pyrometr co nejblíže měřenému objektu, aby viditelná plocha tohoto objektu zcela vyplňovala zorné pole snímače [5].



Obr. 7) Měřený objekt musí zcela vyplňovat měřenou plochu [5]

## 4 ŘÍDÍCÍ JEDNOTKA

Řídící jednotka je nejdůležitější komponentou každého robota. Jedná se o elektronický výpočetní systém, který typicky řídí pohon robota a komunikuje s množinou dostupných senzorů. Pro řízení mobilních robotů se v naprosté většině případů používá tzv. mikrokontrolér, též MCU (z angl. Microcontroller unit).

### 4.1 Mikrokontrolér

Jedná se o elektronickou součástku v podobě integrovaného obvodu s velmi složitou vnitřní křemíkovou strukturou. Mikrokontrolér je schopen sekvenčně vykonávat posloupnost požadovaných instrukcí, které jsou uloženy v interní paměti. Posloupnost instrukcí se zapisuje v některém z programovacích jazyků a pomocí hardwarového programátoru se zavede do paměti MCU. Pro zápis zdrojového kódu s požadovanou funkcionalitou se používá především jazyk C a jazyk Assembler pro některé nízkoúrovňové rutiny. Základní komponenty MCU jsou:

- Procesor neboli CPU (z angl.. Central Processing Unit)
- Paměti pro data a program
- Další specifické periferie jakou jsou např.:
  - Obvody pro správu napájení
  - Obvody pro generování a distribuci hodinového signálu
  - Časovače
  - A/D a D/A převodníky
  - Komunikační rozhraní (UART, I2C, CAN, Ethernet, USB, apod.)

#### 4.1.1 Základní parametry MCU

Při výběru mikrokontroléru se lze nejčastěji setkat s následujícími parametry:

- **Provozní frekvence** – maximální frekvence hodinového signálu jakým je taktováno procesorové jádro. Od provozní frekvence, architektury a instrukční sady procesoru se odvíjí celkový výkon procesoru, vyjadřovaný jako maximální počet operací proveditelných za jednotku času. Typické frekvence pro mikrokontroléry se pohybují v řádech jednotek až stovek MHz.
- **Šířka slova** – je vyjadřována v bitech a definuje maximální velikost operandů, jakou je procesor schopen zpracovat v rámci jedné atomické operace. Od tohoto parametru je odvozena efektivita vykonávání kódu (tedy i rychlost) a také např. velikost adresovatelné paměti. V dnešní době jsou nejpoužívanějšími 16b MCU pro aplikace nevyžadující příliš vysoký výkon. Pro náročnější aplikace pak moderní 32b nebo 64b MCU. Z historického hlediska lze zmínit i 4b a 8b MCU.
- **Velikost a paměti** – velikost interních FLASH a RAM pamětí pro uchování programu a dat.
- **Velikost napájecího napětí**
- **Napětíové úrovně na I/O pinech**
- **Počet I/O pinů**
- **Konfigurace vnitřních periférií:**
  - Typy a počet jednotlivých periférií
  - Rozlišení A/D, D/A převodníků
  - Rozlišení časovačů



## 5 POHONNÝ SUBSYSTÉM ROBOTA

Pohon je nedílnou součástí každého robota. V robotice se pro pohony nejčastěji používají rotační elektromotory. Z této oblasti motorů se pro účely pozemních robotů kategorie hobby nejvíce uplatňují stejnosměrné komutátorové motory, krokové motory a servo motory. V následujících kapitolách bude stručně popsán princip těchto motorů a způsoby jejich řízení.

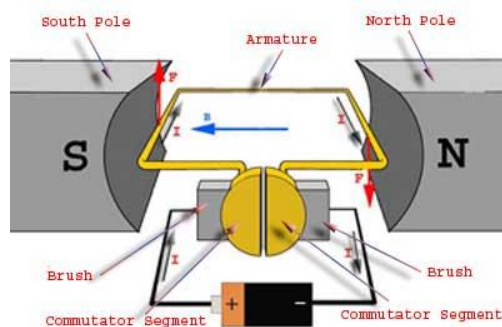
### 5.1 DC motory

Stejnosměrný motor, často označovaný také jako DC motor (z angl. Direct Current) je nejčastěji používaný typ motoru pro pohon mobilních robotů. Jeho časté nasazení vyplývá z několika výhod, kterými jsou např. snadná dostupnost, příznivý poměr výkon/hmotnost, cena či jednoduchý způsob řízení otáček. Ve srovnání s krokovými motory však jeho použití přináší i řadu nevýhod, jako jsou např. obtížnější polohové řízení, nepříjemné elektromagnetické rušení, většinou velmi vysoké otáčky a malý moment. Z důvodu příliš vysokých otáček a malého momentu musí být motor většinou doplněn převodovkou [1].

#### 5.1.1 Princip

Princip stejnosměrného motoru, spočívá v silovém účinku magnetického pole na vodič, kterým prochází elektrický proud. Konstrukčně lze motor rozdělit na dvě části- stator a rotor. Stator je nehybná část motoru, která vytváří díky permanentním magnetům magnetické pole. K němu je připevněno sběrací ústrojí tzv. komutátoru. Rotor neboli kotva je pohyblivá část motoru, která je tvořena hřídelí, vinutím a měděnými, vzájemně izolovanými lamelami komutátoru. Na lamely komutátoru dosedají uhlíkové kartáče. Přes komutátor je potom přiváděn elektrický proud do rotorového vinutí.

Po přivedení stejnosměrného proudu na rotorové vinutí, vznikne kolem tohoto vinutí magnetické pole. Vzájemným působením magnetického pole statoru a rotoru vzniká magnetická tažná síla, která působí na vodiče vinutí kotvy. Směr působení této síly se odvíjí od smyslu průchodu proudu vinutím a smyslu indukčních siločar magnetického pole statoru. Směr působení síly je možné určit Flemingovým pravidlem levé ruky. Aby bylo možno dosáhnout kontinuálního otáčení motoru, je třeba zajistit, aby vodičem kotvy pod severním pólem procházel proud jedním směrem a pod jižním pólem tím stejným vodičem pak směrem opačným. Tento proces změny smyslu protékajícího proudu, který je označován jako komutace, zajišťuje právě komutátor. Komutace DC motoru zajišťovaná mechanickými kontakty je bohužel doprovázena jiskřením a elektromagnetickým rušením. Proto jsou tyto motory pro některé aplikace nevhodné. Čerpáno z [2].



Obr. 8) Elementární DC motor [18]

### 5.1.2 Řízení otáček

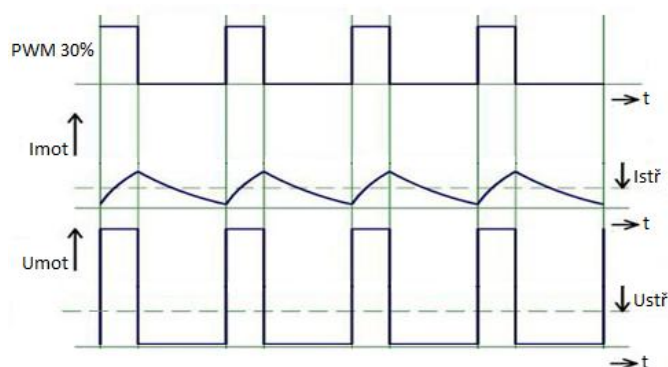
Regulaci otáček DC motoru, resp. příkonu dodávaného do motoru je možné provést podle dvou základních principů[15]:

- **Ztrátové řízení** – řízení otáček je realizováno pomocí spojitě regulace napětí. Konstantní napětí napájecího zdroje je rozděleno odporovým děličem v určitém poměru. Část napětí je nutné na děliči srazit a na motor se pak přivádí napětí ponížené o tento úbytek. Výhodou tohoto řešení je jeho jednoduchost. Používá se pro nenáročné aplikace a motory velmi malých výkonů. Značnou nevýhodou je však fakt, že část výkonu dodávaného zdrojem je děličem transformováno na Joulovo teplo. Při řízení motoru s velkým jmenovitým výkonem vzniká velké množství tepla a nastává pak problém s jeho odvodem. V případě pohonu robota napájeného akumulátorem, lze toto řešení označit za velmi nehospodárné.
- **Bezztrátové řízení** – pro regulaci příkonu dodávaného do motoru se používá některá z pulzních modulací. Nejčastěji pulzně-šířková modulace neboli PWM (Pulse-Width Modulation).

#### Pulzně šířková modulace (PWM)

Při regulaci otáček pulzně šířkovou modulací se na rozdíl od ztrátového řízení nemění úroveň napětí na motoru, ale mění se doba, po kterou je motor k napájecímu napětí připojen. Konstantní napájecí napětí se tedy v čase rozloží do sledu obdélníkových impulzů s konstantní frekvencí a s proměnnou šířkou. Poměr dob trvání aktivní a klidové úrovně PWM signálu se označuje jako střída a vyjadřuje se v rozsahu 0 až 100%.

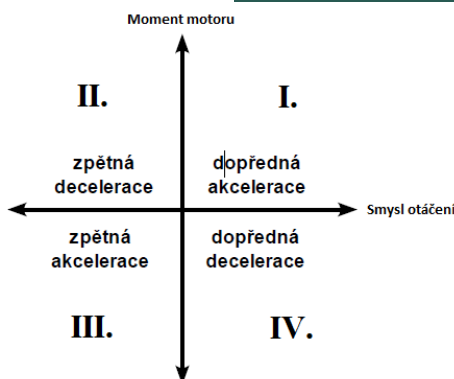
Motor buzený PWM signálem s konkrétní střídou se chová jako by byl spojitě napájen střední hodnotou napětí tohoto signálu. Jestliže je frekvence PWM dostatečně vysoká, tak indukčnost vinutí motoru způsobí, že proud tekoucím vinutím nestačí sledovat rychlé změny napájecího napětí a průběh proudu v čase je pak více nebo méně zvlněný viz ilustrace níže. Použitím PWM lze dosáhnout kvalitní plynulé regulace otáček s velmi malými ztrátami na výkonových prvcích, neboť tyto pracují ve spínacím režimu [1].



Obr. 9) Průběh PWM signálu se střídou 30% a průběhy napětí a proudu na motoru

### 5.1.3 Řízení smyslu otáček a vhodné budící obvody

Kromě požadavku na řízení otáček, je u robota nutné řídit i smysl otáček. Stejnosměrný motor lze provozovat v jednom ze 4 pracovních režimů (kvadrantů). Možné kvadranty jsou zobrazeny na obrázku níže. Informace týkající se způsobů buzení motorů byly čerpány ze zdroje [15].



Obr. 10) Pracovní režimy stejnosměrného motoru [15]

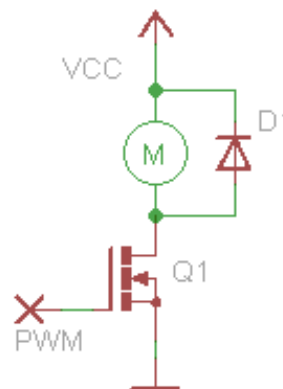
Podle toho v jakých kvadrantech bude stejnosměrný motor řízen, je nutné zvolit vhodný budicí obvod. Běžně se lze v praxi setkat se třemi způsoby řízení, přičemž každé se liší oblastí použitelnosti a složitostí zapojení budicího obvodu.

#### Jednokvadrantové řízení

Tento způsob řízení umožňuje řídit motor pouze v jednom směru (kvadranty I a III) a neumožňuje brždění. Toto řízení rovněž neumožňuje reverzaci otáček, a proto není tento způsob buzení pro pohon robota vhodný.

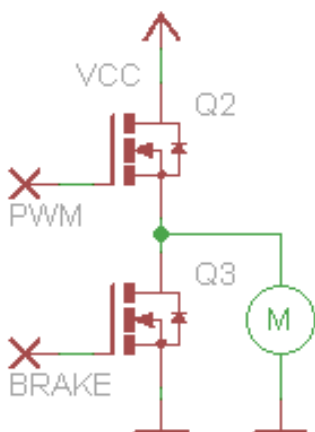
Tranzistor Q1 pracuje ve spínacím režimu. V okamžiku otevření tranzistoru začne přes motor téct proud a motor se roztočí směrem odpovídajícím smyslu protékajícího proudu.

Jelikož se motor chová jako indukční zátěž, na které se při skokovém uzavření tranzistoru objevují velké proudové špičky, je celé zapojení doplněno o ochrannou diodu zabráňující proražení tranzistoru. V okamžiku uzavření tranzistoru se proud přes diodu přeměruje zpět do motoru a tím se proudová špička postupně zmaří.



Obr. 11) Jednokvadrantový budič

#### Dvoukvadrantové řízení



Obr. 12) Dvoukvadrantový budič

Dvoukvadrantové řízení vzniká z jednokvadrantového přidáním dalšího tranzistoru, který slouží k brždění motoru v případě, že motor přejde do režimu generátoru (motor se točí, ale není buzen). Motor lze řídit v kvadrantech I a IV nebo II a III. Ani dvoukvadrantový budič neumožňuje snadnou reverzaci otáček a není tedy pro pohon robota vhodný.

Při použití tohoto budiče je třeba zamezit současnému sepnutí obou tranzistorů, protože v takovém případě vzniká mezi oběma póly napájecího zdroje zkrat.



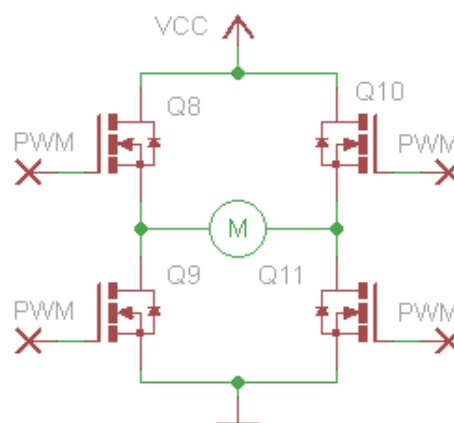
Čtyřkvadrantové řízení

Přidáním druhého páru tranzistorů ke dvoukvadrantovému budiči vzniká můstkové zapojení umožňující snadné řízení motoru ve všech kvadrantech. Takové zapojení se potom nazývá čtyřkvadrantové nebo častěji tzv. H-můstek. Tímto zapojením je možno elektronicky měnit polaritu napětí na motoru a tím obracet smysl otáčení.

Pro chod motoru v jednom i druhém směru, je třeba vždy sepnout dvojici tranzistorů Q8+Q11 nebo Q9+Q10, což lze oproti předchozím způsobům řízení považovat za nevýhodu. Navíc je opět nutné pamatovat na nebezpečí zkratu napájecího zdroje.

Celou situaci je však možné efektivně řešit použitím H-můstku ve formě integrovaného obvodu, který již v sobě má zaintegrovánu veškerou logiku pro spínání vnitřních tranzistorů a navíc redukuje počet signálů potřebných pro řízení motoru v obou směrech.

Takový H-můstek pak má jeden vstup pro PWM signál a druhý vstup pro určení požadovaného směru otáčení. Jedna logická úroveň signálu pak představuje chod vpřed a druhá chod vzad.



Obr. 13) Čtyřkvadrantový budič

## 5.2 Krokové motory

Na rozdíl od stejnosměrného motoru, který v celém rozsahu otáček vykazuje spojitý pohyb výstupní hřídele, je pohyb výstupní hřídele krokového motoru při nízkých otáčkách nespojitý a rotor se pohybuje v tzv. krocích mezi stabilními polohami. Při vyšších otáčkách už krokování nelze postřehnout a hřídel vykazuje spojitý pohyb. S krokovými motory se lze v robotice též velmi často setkat, právě kvůli jejich schopnosti krokování. Ta umožňuje velmi snadné řízení otáček i polohy.

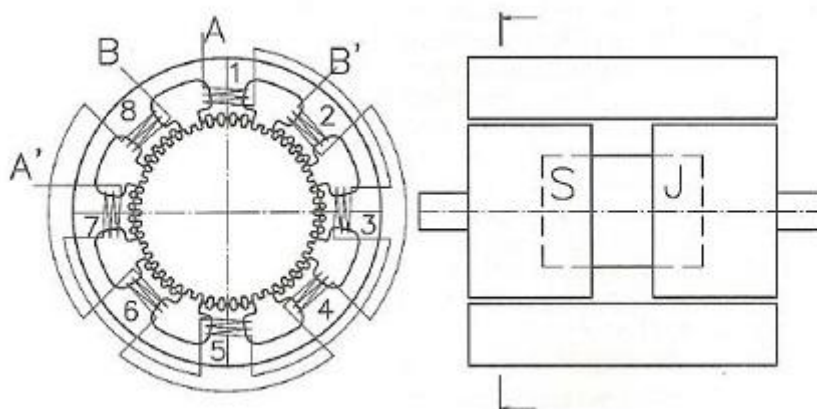
### 5.2.1 Rozdělení krokových motorů podle konstrukce rotoru

Krokový motor (KM) je stejně jako stejnosměrný motor tvořen statorem a rotorem. Stator tvoří ocelové plechy s pólovými nástavci, na který jsou navinuty fáze statorového vinutí. Dle konstrukce rotoru se pak krokové motory rozlišují na motory [8]:

- **s pasivním rotorem** – u tohoto typu je rotor tvořen kovovým profilem nebo svazkem železných plechů z magneticky měkkého materiálu nalisovaného na hřídeli.
- **s aktivním rotorem** – rotor se skládá z hřídele a permanentních magnetů. Dle orientace permanentních magnetů lze pak motory dále dělit na motory radiálně nebo axiálně polarizované.
- **hybridní** – rotor tohoto typu motoru kombinuje obě výše zmíněné konstrukce. Je tvořen kovovým profilem nebo svazkem železných plechů s vystupujícími pólovými nástavci a mezi tyto nástavce jsou vloženy permanentní magnety. Na nástavcích rotoru i statoru je vytvořeno jemné drážkování, které zvyšuje působení magnetického toku ve vzduchové mezeře mezi rotorem a statorem. Tento typ motoru je v dnešní době nejvíce používán, neboť vykazuje lepší statické a dynamické vlastnosti oproti oběma



předcházejícím typům.



Obr. 14) Řez hybridním KM [1]

### 5.2.2 Princip

Pro fungování krokového motoru je nutné, aby rotor měl menší počet pólových párů než stator. Tím pádem nikdy nemůže nastat situace přesného sesouhlasení všech nástavců statoru i rotoru (případně jejich drážkování). Po přivedení stejnosměrného impulsu na některou z fází statorového vinutí dojde k vyvolání magnetického pole, které silově působí na rotor.

V případě, že je impuls přiveden na fázi, která je navinuta na přesně sesouhlaseném pólovém nástavci statoru vůči rotoru, nedojde k žádnému pohybu a rotor bude ve stávající pozici přidržován až do okamžiku odeznění budicího impulsu.

V případě, že pólové nástavce sesouhlaseny nejsou a budicí impuls bude přiveden na vhodnou fázi, dojde působením vyvolaného magnetického pole k natočení rotoru do takové pozice, kdy je dosaženo minimálního magnetického odporu. Tedy do pozice, kdy jsou nástavce rotoru i statoru v jedné ose. Tímto motor provedl jeden krok. Velikost kroku motoru je závislá na počtu pólových párů a v případě hybridních motorů i na jemnosti drážkování.

### 5.2.3 Řízení krokových motorů

Řízení krokového motoru je velmi jednoduché. Je založeno na vhodné sekvenci spínání jednotlivých fází statorového vinutí. Prvním krokem je tzv. zapolohování – natočení rotoru do definované výchozí pozice. Toho se docílí sepnutím libovolné fáze a od této je pak třeba vycházet při spínání dalších fází během rozběhu motoru.

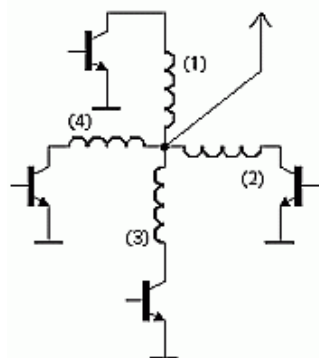
Pro řízení polohy je třeba znát úhlovou velikost jednoho kroku a v elektronice, která řídí krokování potom počítat, kolik kroků již bylo provedeno od výchozí pozice. Rychlost otáčení se řídí změnou frekvence spínání jednotlivých fází.

Možnost otáčet rotorem po krocích umožňuje řízení rychlosti i polohy v otevřené smyčce bez jakékoliv zpětné vazby. To sebou však nese určitá úskalí a motor je třeba dostatečně dimenzovat. Nesmí totiž dojít k situaci, kdy frekvence krokování přesáhne maximální možnou frekvenci pro daný zátěžný moment na hřídeli. Při překročení této hranice dojde k výpadku motoru ze synchronních otáček a ztrátě kroků. To sebou logicky přináší i ztrátu informace o poloze rotoru. Je však samozřejmě možné i řízení se zpětnou vazbou.

Řízení krokového motoru z pohledu spínání jednotlivých fází je možno rozdělit na několik způsobů, přičemž některé způsoby lze mezi sebou kombinovat, čímž lze dosáhnout například vyššího momentu či přesnějšího krokování. Způsoby řízení čerpány z [9], [8].

Unipolární řízení

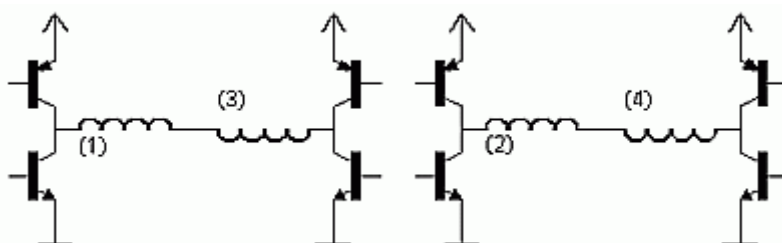
Unipolární řízení je charakteristické tím, že proud prochází v jednom časovém okamžiku pouze jednou cívkou. Tento druh řízení s sebou přináší výhody i nevýhody. Nevýhodnou je menší krouticí moment. Ten je však vyvážen nízkým příkonem a jednoduchou řídicí elektronikou, kdy pro ovládání motoru postačuje jeden spínací prvek (většinou tranzistor) na každou cívku. Unipolární řízení je možné pouze u unipolárních KM s odpovídajícím vnitřním zapojením statorového vinutí.



Obr. 15) Unipolární řízení [9]

Bipolární řízení

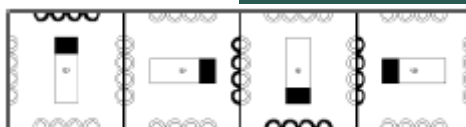
V případě bipolárního řízení prochází proud v jednom časovém okamžiku vždy dvěma protilehlými sériově zapojenými cívkami. Při srovnání s unipolárním způsobem řízení je možné vyvinout dvojnásobný krouticí moment za cenu většího proudového odběru. Nevýhodou je nutnost použití složitější výkonové elektroniky, kdy jsou pro řízení třeba dva H-můstky. Bipolární řízení je možné pouze u bipolárních KM s odpovídajícím vnitřním zapojením statorového vinutí.



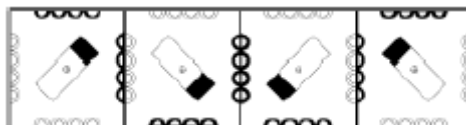
Obr. 16) Bipolární řízení [9]

Jednofázové a dvoufázové řízení

Při jednofázovém řízení je v jednom časovém okamžiku aktivní vždy pouze jedna cívka, u dvoufázového řízení jsou aktivní vždy cívkové dvě. U obou způsobů řízení dochází k provedení stejného počtu kroků v rámci jedné mechanické otáčky, ale motor řízený dvoufázově dosahuje dvojnásobného krouticího momentu, přičemž však spotřebuje dvojnásobně více elektrické energie ve srovnání s motorem řízeným jednofázově. Oba způsoby řízení jsou někdy označovány jako čtyřtaktní. Princip je možné si prohlédnout na obrázcích níže. Každý kvadrant obrázku odpovídá jednomu stavu motoru s vyznačenou aktivní fází a adekvátně natočeným rotorem.



Obr. 17) Jednofázové řízení [9]

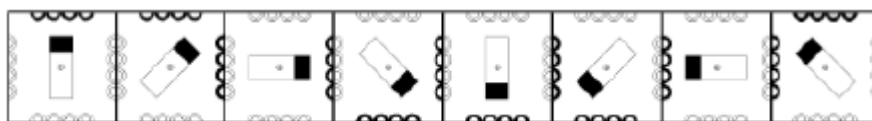


Obr. 18) Dvoufázové řízení [9]

### Řízení s polovičním krokem

Při řízení s polovičním krokem neboli osmitaktním řízení se střídavě spíná jedna a následně dvě sousední fáze motoru. Jedná se vlastně o kombinaci jednofázového a dvoufázového řízení. Tímto způsobem lze dosáhnout dvojnásobného počtu kroků v rámci jedné mechanické otáčky oproti řízení jednofázovému nebo dvoufázovému. Rotor tedy lze polohovat s dvojnásobným rozlišením.

V návrhu pohonu s osmitaktně řízeným krokovým motorem je nutné počítat s tím, že přídržný moment závisí na počtu aktivních vinutí motoru. V okamžiku kdy je sepnuta pouze jedna fáze, je přídržný moment poloviční.



Obr. 19) Princip řízení s polovičním krokem [9]

### Mikrokrokování

Metoda mikrokrokování se velmi podobá dvoufázovému řízení, kdy jsou v jednom okamžiku sepnuté vždy dvě sousední fáze motoru. U mikrokrokování se však navíc řídí příkon dodávaný do obou fází. Příkon lze rozdělovat mezi obě fáze v libovolném poměru a tomuto poměru pak odpovídá velikost rotorového kroku. Výhodou tohoto řešení je možnost dosáhnout velikosti kroku v řádech desetin až setin úhlových stupňů.

## 5.3 Servomotory

Servomotor (servo) je speciální druh DC motoru, který dokáže otočit hřídelí do požadované pozice a tuto pozici pak drží. Natočení hřídele je typicky možné pouze v určitém rozsahu, např.  $0^\circ$  až  $180^\circ$ . Jsou však i serva, která se dokáží otáčet kontinuálně. Servo kromě vlastního motoru obsahuje převodovku a malý potenciometr spřažený s výstupní hřídelí. Tento potenciometr slouží jako zpětná vazba pro elektronický obvod regulující požadovaný úhel natočení. Nedostatkem těchto servomotorů je mechanické opotřebování potenciometru, proto se u kvalitnějších motorů pro zpětnou vazbu používají např. rotační enkodéry [1].

### 5.3.1 Řízení servomotoru

Pro nastavení žádané hodnoty úhlu natočení je servomotor vybaven analogovým vstupem, na který se přivádí širkově modulovaný obdélníkový signál. Perioda tohoto signálu je typicky 20 ms a šířka pulzu v rozsahu 1 až 2 ms. Šířka pulzu 1 ms pak odpovídá jedné krajní poloze, 2

ms druhé krajní poloze a 1,5 ms pak střední poloze. Vhodnou délkou pulzu tedy lze nastavit libovolnou polohu výstupní hřídele serva [1].



Obr. 20) Modelářské servo [1]

## 6 SENZORICKÝ SUBSYSTÉM ROBOTA

Senzory jsou důležitou součástí každého robota. Umožňují interakci robota s okolím, detekci překážek, sledování polohy robota či měření provozních parametrů pro uzavření zpětných vazeb regulačních obvodů. Senzory lze klasifikovat podle mnoha měřítek, například podle měřené veličiny nebo fyzikálního principu. Z hlediska vztahu senzoru k robotu a okolí lze rozlišovat [1]:

- **Interní senzory** – za interní senzory jsou považovány ty, které měří provozní a diagnostické parametry robota. Tím se rozumí, že tyto senzory poskytují informaci například o stavu pohonného systému (rychlost, natočení) nebo relativní pozici v souřadném systému, dále pak informace o teplotě robota či stavu baterie. Typickými představiteli této kategorie jsou rotační enkodéry, gyroskopy, akcelerometry apod.
- **Externí senzory** – jako externí jsou označovány senzory, pomocí kterých robot vnímá svoje okolí. Dle způsobu měření lze dále tuto skupinu rozdělit na dotykové a bezdotykové. Dotykové umožňují měření pouze při fyzickém kontaktu, kdežto bezdotykové senzory pro měření využívají různé fyzikální jevy, které různě ovlivňují senzor. Do této kategorie patří například detektory překážek, dálkoměry, laserové radary, senzory teploty či kamery.

### 6.1 Rotační enkodéry

Při řízení mobilního robota je třeba velmi často znát jeho polohu, vzdálenost jakou robot ujel nebo rychlost jeho pohybu. U pozemních kolových či pásových robotů se k měření těchto veličin využívá tzv. rotačních enkodérů. V následující kapitole budou představeny typy a principy enkodérů, s nimiž se lze v robotice setkat nejčastěji. Čerpáno z [1].

#### 6.1.1 Fyzikální principy a detekční schopnosti rotačních enkodérů

Rotační enkodér je druh snímače, který převádí rotační pohyb na elektronický signál. V robotice se lze nejčastěji setkat s enkodéry pracujícími na optickém nebo magnetickém principu. Podle detekčních schopností se pak enkodéry dělí na enkodéry absolutní a přírůstkové (inkrementální).

##### Optické enkodéry

Optické enkodéry jsou nejčastěji realizované jako průchozí nebo odrazové optické závory. V obou případech enkodér obsahuje clonu, zdroj světla a světlocitlivý snímač. Clona je mechanicky spojena s otáčejícím se kolem nebo hřídelí a obsahuje posloupnost průhledných a neprůhledných otvorů, či odrazných a neodrazných plošek. Otáčející se clona propouští nebo odráží světelné impulzy. Tyto impulzy aktivují světlocitlivý snímač, který je převádí s pomocí další elektroniky (tvarovací obvody) na obdélníkový signál.

##### Magnetické enkodéry

Magnetické enkodéry pracují na principu Hallovy sondy, kdy snímač detekuje změny magnetického pole vyvolaného pohybem magnetického kotoučku s určitým počtem pólových párů. Výstupem tohoto enkodéru je opět obdélníkový signál, jehož stavy se mění na rozhraní dvou oblastí kotoučku s opačnou magnetickou polaritou.

V současné době se na trhu také objevují magnetické enkodéry ve formě integrovaného obvodu, který zapouzdřuje funkce Hallovy sondy, analogového rozhraní a

digitálního zpracování signálu. Obvykle lze u nich programově konfigurovat přesnost nebo volit mezi inkrementálním, absolutním či PWM výstupním režimem. Komunikace s takovýmto enkodérem pak probíhá prostřednictvím vhodné sběrnice, např. SSI.



Obr. 21) Magnetický enkodér v integrovaném provedení [21]

### Absolutní enkodéry

Tento typ enkodéru poskytuje absolutní informaci o natočení v rozsahu 0 až 360°. Pro větší počet otáček je vybaven výstupem, jehož hodnota se s každou otáčkou zvýší nebo sníží, dle směru otáčení. Znalost absolutní pozice je vykoupena složitější konstrukcí a principem kódování. Absolutní enkodéry pracují na optickém principu, který je popsán výše.

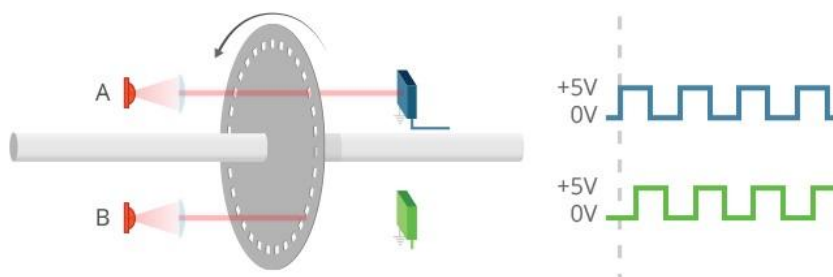
Kódování absolutní pozice je možné díky speciálnímu optickému kotouči, který obsahuje určitý kódový vzor. Kotouč je z jedné strany prosvětlován a z druhé strany kotouče je světlocitlivý snímač. Tento snímač reaguje na světlo, jenž prošlo skrz kotouč. Výstupem je pak binární informace o absolutní poloze. Výhodou je, že tento typ enkodéru nevyžaduje referenční bod po zapnutí systému. Jakmile je enkodér aktivován, okamžitě na svém výstupu nastaví absolutní informaci o natočení. Nevýhodou je pak větší složitost a cena.



Obr. 22) Absolutní enkodér [22]

### Inkrementální enkodéry

Inkrementální enkodér generuje dva vzájemně fázově posunuté signály<sup>1</sup> označované nejčastěji A, B. Podle rychlosti změn úrovní jednoho ze signálů lze odměřit rychlost otáčení a podle počtu změn úhel natočení vzhledem k referenční pozici. Pokud je sledována posloupnost změn úrovní na obou signálech současně, resp. jejich fázi, lze určit i směr otáčení.



Obr. 23) Kvadraturní signál generovaný inkrementálním enkodérem

#### **6.1.2 Kvadraturní dekódování**

Kvadraturní signál generovaný inkrementálním enkodérem je možné zpracovávat hardwarově nebo softwarově. Obecně nelze říci, která metoda je lepší, vždy je třeba vzít do úvahy frekvenci detekovaného signálu a to, co z něho bude zjišťováno. Čerpáno z [16].

Výhodou hardwarového zpracování je nízká výpočetní zátěž nadřazeného systému. Tato vlastnost se pak uplatňuje především pro velmi rychlé signály. Nevýhodou je potřeba další elektroniky a specializovaných obvodů s funkcí kvadraturního dekóderu.

Výhodou softwarového zpracování je nižší obvodová složitost, jelikož celé vyhodnocování lze provádět pomocí vhodně naprogramovaného MCU. Některé mikrokontroléry určené pro použití v oblasti řízení motorů přímo obsahují periférii pro kvadraturní dekódování. Tam, kde tato periférie dostupná není, je možné dekódování řešit pomocí přerušení MCU, které je vyvoláno nástupnou, sestupnou, případně oběma hranami kvadraturního signálu. U tohoto řešení je ale třeba pamatovat na to, že obsluha přerušení musí být co nejkratší, aby bylo přerušení dokončeno s dostatečným předstihem před příchodem dalšího přerušení a zároveň zbyl dostatek výpočetního času pro ostatní funkce MCU. Za nevýhodu tohoto řešení lze považovat variabilní zátěž MCU, která roste s frekvencí dekódovaného signálu.

Variabilní zátěž lze eliminovat, jestliže místo asynchronně vyvolávaných přerušení bude kvadraturní signál synchronně vzorkován. Toho se docílí periodickým vyvoláním přerušení časovače, v jehož obsluze lze pak jednoduše sledovat úrovně (stavy) kvadraturního signálu. Frekvenci, s jakou bude toto přerušení vyvoláváno, je třeba volit tak, aby byla alespoň 4x rychlejší, než jaká je maximální možná frekvence sledovaného signálu. Zátěž MCU je pak vyšší, ale za to konstantní.

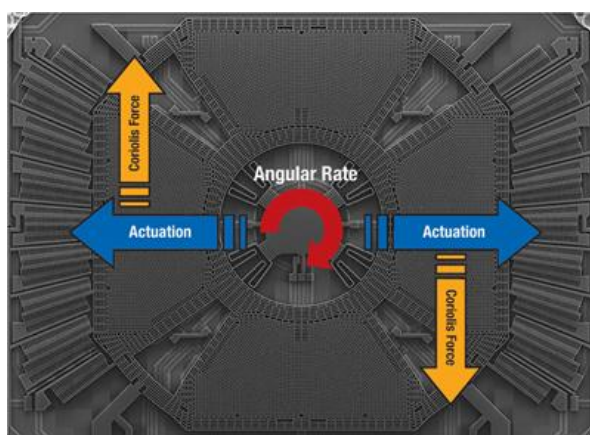
<sup>1</sup> V ideálním případě obdélníkového tvaru. Prakticky však mají signály často sinusový průběh a pro kvadraturní dekódování je tedy nutné je upravit vhodným tvarovacím obvodem.



## 6.2 MEMS gyroskopy

MEMS gyroskop je elektronická součástka v integrovaném provedení, sloužící pro měření úhlového zrychlení podle dané osy. Základem je mechanicky rezonující (vibrující) hmota o přesně definované hmotnosti. Je-li směr rezonance kolmý na osu, podle které dojde k pootočení či naklopení gyroskopu, tak vlivem tohoto pootočení začne na hmotu působit virtuální Coriolisova síla viz obrázek 24. Na obrázku je smysl rotace vyznačen červenou šipkou, směr rezonance hmoty modrými šipkami a směr působení Coriolisovy síly žlutými šipkami.

Vlivem účinku Coriolisovy síly dojde k vychýlení rezonující hmoty z výchozí pozice a vzájemnému posunutí měřicích plošek, které představují elektrody vzduchových kondenzátorů. Posuvem těchto plošek se úměrně mění kapacita. Vnitřní obvody gyroskopu pak převádějí kapacitu na hodnotu aktuální úhlové rychlosti. [20].



Obr. 24) Struktura MEMS gyroskopu [23]

### 6.2.1 Vyhodnocení úhlu natočení

Výstupem MEMS gyroskopů je nejčastěji digitální údaj o úhlové rychlosti v jednotkách DPS (z angl. Degrees Per Second). Aktuální úhel natočení v úhlových stupních je možné získat integrací úhlové rychlosti v čase:

$$\theta = \int \omega \cdot dt \quad (11)$$

Kde  $\theta$  je úhel natočení [ $^{\circ}$ ] a  $\omega$  je úhlová rychlost [ $1^{\circ} \cdot s^{-1}$ ].

Integrace úhlové rychlosti v čase s sebou však nese nepříjemný problém označovaný jako gyroskopický drift. U gyroskopu, který je v naprostém klidu, by se dalo očekávat, že hodnota aktuální úhlové rychlosti bude vždy rovna nule a tedy i úhel natočení bude roven nule. Bohužel tomu tak ale není. MEMS gyroskop je negativně ovlivňován teplotou a hlavně pak povrchovým napětím, které vzniká pouhým připojením gyroskopu na DPS.

Tyto vnější vlivy pak mají za následek, že data získávaná z gyroskopu vykazují určitý offset a integraci v čase pak logicky vzniká značná chyba měření. Problém driftu je možné z velké části eliminovat kalibrací. Z gyroskopu, který je v naprostém klidu, se přečte před „ostrým“ měřením např. 100 hodnot, a z těchto se pak spočte aritmetický průměr. Vypočtená hodnota je považována za stálý offset gyroskopu a vždy se tato hodnota přičte nebo odečte (dle znaménka) k nově změřené aktuální hodnotě úhlové rychlosti. Tím se problém driftu z velké části eliminuje, ale bohužel úplně nezmizí.



Dalšího zvýšení přesnosti měření lze dosáhnout, použije-li se vedle gyroskopu i akcelerometr. Gyroskop je vhodný pro měření rychlých změn natočení, kdežto akcelerometr poskytuje přesné data spíše pro pomalé děje. Data z gyroskopu a akcelerometru lze pak zkombinovat komplementárním či složitějším Kalmanovým filtrem a dosáhnout tak daleko věrohodnějších výsledků.

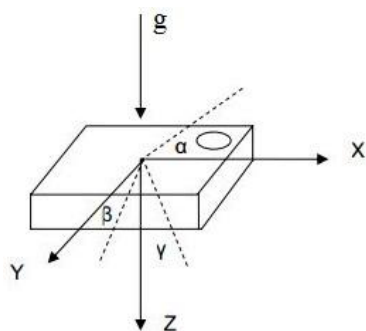
### 6.3 MEMS akcelerometry

Akcelerometry slouží pro měření dynamického a statického zrychlení, jenž na akcelerometr působí v dané ose. Stejně jako gyroskop může být použit i pro měření náklonu. V dnešní době jsou hojně využívány akcelerometry v provedení MEMS s proměnnou kapacitou. Výhodou MEMS akcelerometrů je možnost měření ve všech třech osách při použití pouze jedné součástky.

Základem akcelerometru je stejně jako u gyroskopu seismická hmota vytvořená v elektro-mechanické křemíkové struktuře. Při působení zrychlení je hmota vychylována z výchozí pozice. To vede k vzájemnému posuvu elektrod vzduchových kondenzátorů a ke změně výsledné kapacity.

#### 6.3.1 Vyhodnocení úhlu náklonu

Ve statické poloze na akcelerometr působí pouze gravitační zrychlení o velikosti 1 G ( $9,806 \text{ m}\cdot\text{s}^{-2}$ ). Je-li akcelerometr orientován na plochu, kdy osy X a Y v Kartézském souřadném systému jsou rovnoběžné se zemským povrchem, tak vektor gravitačního zrychlení je na tyto dvě osy kolmý. Účinky tohoto zrychlení jak pak možné naměřit pouze v ose Z. Při náklonu akcelerometru začnou účinky gravitačního zrychlení v ose Z klesat a naopak narůstat v osách X nebo Y. Úhel náklonu je pak možné vyjádřit pomocí trigonometrických funkcí:



Obr. 25) Měření náklonu [25]

$$\alpha = \arcsin \frac{A_x}{g} \quad (12)$$

$$\beta = \arcsin \frac{A_y}{g} \quad (13)$$

$$\gamma = \arccos \frac{A_z}{g} \quad (14)$$

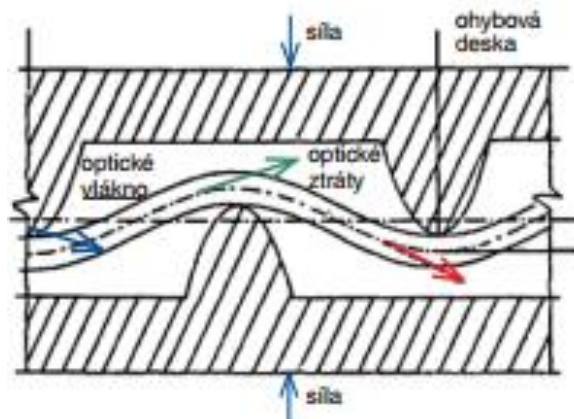
Kde  $\alpha$ ,  $\beta$ ,  $\gamma$  jsou úhly náklonu  $[\circ]$ ,  $A_x$ ,  $A_y$ ,  $A_z$  je naměřené zrychlení v daných osách  $[\text{m}\cdot\text{s}^{-2}]$  a  $g = 9,806 \text{ m}\cdot\text{s}^{-2}$  je gravitační zrychlení.

### 6.4 Taktilní senzory

Taktilní senzor se uplatňuje v případech, kdy je třeba detekovat a snímat dotyk s předměty z okolí robota. Nejjednodušším příkladem taktilního senzoru je mechanický kontaktní spínač aktivovaný při kontaktu s překážkou. Při nárazu na překážku dojde k sepnutí, nebo rozepnutí elektrického obvodu a tyto stavy jsou dále vyhodnocovány řídicí jednotkou.

Existují však mnohem sofistikovanější senzory nahrazující lidský hmat, které měří úchopnou sílu nebo prokluzování uchopeného předmětu. Z fyzikálního principu lze rozlišovat [26]:

- **Elastomerové senzory** – využívají změny odporu vodivého elastomeru k převodu síly a tlaku na elektrický signál. Z pohledu konstrukce jde o silikonovou pryž sycenou grafitem nebo kovovým prachem. Při stlačení se pak mění uspořádání těchto dopovaných částic a mění se odpor mezi vyvedenými elektrodami.
- **Tenzometrické senzory** – používají se například pro měření síly působící na předmět při jeho uchopení robotem. Jako převodník síly se uplatňuje například polovodičový tenzometr připevněný přímo na uchopovaný předmět.
- **Kapacitní senzory** – využívají změnu velikosti společných ploch válcových nebo deskových kondenzátorů.
- **Piezoelektrické** – pro převod síly na elektrický signál je využíváno piezoelektrických materiálů. Nevýhodou těchto snímačů je, že nesmí být proudově zatěžovány a vykazují větší nároky na snímací elektroniku. Pro snímání se používají napěťové zesilovače s velmi velkým vstupním odporem, řádově  $1 \cdot 10^{12} \Omega$ .
- **Optické senzory** – jsou tvořeny optickými vlákny. Při měření se využívá mikroohybu vláken, při kterém se mění optické podmínky pro průchod světla od zdroje (laserová dioda, LED) k detektoru. Mechanický tlak na senzor je pak detekován jako změna intenzity, fáze nebo spektrálních vlastností procházejících světelných vidů. Výhodou optických senzorů je jejich necitlivost vůči okolnímu elektromagnetickému záření a možnost přenášet informaci mezi objekty s různým elektrickým potenciálem a to i na velké vzdálenosti.



Obr. 26) Princip optického taktilního senzoru s mikroohybem [26]

## 6.5 Sonary

Sonar se využívá pro detekci překážek a měření vzdáleností. Při měření sonar vyšle akustický signál a měří dobu, za jakou se k sonaru vrátí odraz tohoto signálu od překážky. Frekvence emitovaného signálu leží mimo slyšitelné pásmo, zpravidla jde o frekvence nad 40kHz. Proto jsou tyto senzory označovány také jako ultrazvukové. Výhodou sonarů je, že rychlost šíření akustického signálu vzduchem je poměrně nízká a tak doba mezi vysláním a přijetím odraženého signálu výrazně vyšší než například u laserových či infračervených detektorů. Proto sonary poskytují vysoké rozlišení a přesnost i pro malé vzdálenosti. Z pomalého šíření zvuku ve vzduchu však plyne nižší frekvence, s jakou lze měření opakovat. Opakovat měření je možné až po přijetí akustického signálu, což může být poměrně dlouhá doba v případě větších vzdáleností překážky od sonaru. Za nevýhodu lze považovat i tlumení ultrazvukového signálu ve vzduchu, díky čemuž klesá maximální dosah sonaru na desítky i jednotky metrů. V případě použití více sonarů současně, nelze provádět všechna měření v jednom čase. Důvodem je fakt, že sonar pak zachytává i signály emitované jiným sonarem. Měření je tedy

nutné provádět sekvenčně a s dostatečnou prodlevou nebo použít sonary emitující signály s různou frekvencí [1].

### 6.5.1 Stanovení vzdálenosti překážky od sonaru

Aby bylo možné stanovit vzdálenost od překážky, je nutné převést dobu mezi vysláním a přijetím akustického signálu v jednotkách času, na jednotky délky. Přitom je třeba si uvědomit, že tato odpovídá dvojnásobku vzdálenosti překážky od sonaru, neboť ultrazvukový signál se nejprve šíří od sonaru k překážce a pak se od překážky odráží zpět. Klíčem pro stanovení vzdálenosti je vztah pro přibližné určení rychlosti šíření zvuku v suchém vzduchu [17]:

$$v = (v_{T0} + 0,607 \cdot T) \quad (15)$$

Kde  $T$  je teplota vzduchu [ $^{\circ}\text{C}$ ] a  $v_{T0} = 331,57 \text{ m}\cdot\text{s}^{-1}$  je rychlost zvuku při  $0^{\circ}\text{C}$ .

Ze vztahu je patrné, že rychlost zvuku ve vzduchu je závislá na teplotě. Pro teplotu vzduchu  $15^{\circ}\text{C}$  činí rychlost zvuku  $340 \text{ m/s}$ . Je-li doba trvání Echo impulzu v tomto prostředí rovna  $10 \text{ ms}$ , pak vzdálenost překážky od sonaru lze určit snadno:

$$s = v \cdot t/2 = 340 \cdot 0,01/2 = 1,7 \text{ m} \quad (16)$$

Kde  $s$  je vzdálenost [ $\text{m}$ ],  $v$  je rychlost zvuku ve vzduchu [ $\text{m}\cdot\text{s}^{-1}$ ] a  $t$  je doba trvání Echo impulzu [ $\text{s}$ ].



## 7 KOMUNIKAČNÍ SUBSYSTÉM ROBOTA

Komunikační rozhraní je velmi důležitou součástí celého robota. Obsluha může jeho prostřednictvím provádět zásah do řízení robota nebo je lze využít pro sběr provozních dat. S výhodou je lze také uplatnit pro ladící účely, kdy lze "za běhu" nastavovat některé provozní parametry (např. koeficienty PID regulátorů apod.) bez nutnosti přeprogramování MCU. Dle fyzického provedení rozlišujeme mezi drátovou a bezdrátovou komunikací. V oblasti mobilních robotů se používají oba přístupy, ale každý k jinému účelu.

Komunikace s obsluhou (stolní PC, notebook či mobilní telefon) probíhá nejčastěji bezdrátově. Řadíme sem např. standardy WiFi, Bluetooth a ZigBee. V robotice se nejčastěji uplatňují poslední dvě jmenované technologie a tyto budou níže podrobněji rozebrány,

Drátová komunikace se používá především pro komunikaci mezi MCU a dalšími obvody, jimiž je robot osazen (senzory, paměti nebo specializované moduly, které převádějí drátovou na bezdrátovou komunikaci). Pro drátovou komunikaci slouží například sběrnice SPI, I2C, USB, Ethernet či asynchronní sériová linka UART a její průmyslové rozšíření RS-232.

### 7.1 Bluetooth

Technologie Bluetooth, definovaná standardem IEEE 802.15.1, je určena výhradně pro komunikaci na krátké vzdálenosti a pracuje v nelicencovaném frekvenčním pásmu 2.4 GHz. Zařízení Bluetooth se vyskytují v několika revizích a výkonových třídách, podle nichž se stanovuje rychlost přenosu dat (až 24 Mbps) a přibližný dosah (1 až 100 metrů). Pomocí Bluetooth je možné propojit až 8 zařízení do specializované lokální sítě Piconet, kde jedno zařízení je typu master a ostatní typu slave. Použití Bluetooth pro mobilní roboty je výhodné kvůli velkému rozšíření tohoto standardu na poli dnešní osobní elektroniky, kterou pak lze využít jako obslužný terminál [29].

### 7.2 ZigBee

Standard ZigBee, označovaný též jako IEEE 802.15.4, je svým využitím orientován především do oblasti průmyslu, kde se uplatňuje jako bezdrátové komunikační rozhraní pro monitorování určitého systému nebo pro komunikaci systému s různými senzory a regulátory. Stejně jako u technologie Bluetooth jsou signály šířeny v nelicencovaném pásmu 2.4 GHz. Ve srovnání s technologií Bluetooth je ZigBee velice pomalé (v Evropě typicky 20 kbps) a po stránce dosahu přibližně srovnatelné řešení. Velkou výhodou ZigBee je však jednoduchý komunikační protokol, nízké HW nároky a minimální spotřeba energie, přičemž ZigBee umožňuje vznik poměrně složitých sítí s vysokou spolehlivostí [30].



## 8 NÁVRH ROBOTA

V následujících kapitolách budou rozebrány jednotlivé komponenty použité po stavbu robota.

### 8.1 Podvozek

Podvozek je základním prvkem robota a svou konstrukcí značnou mírou předurčuje vhodnost robota pro nasazení v té či oné aplikaci. Při výběru podvozku jsem se řídil následujícími elementárními faktory:

- Pro navrhovaného robota použít pokud možno již existující ověřený podvozek s dobrou dostupností náhradních dílů a vyhnout se tak složitému návrhu a výrobě vlastního podvozku.
- Vhodnost podvozku pro zvolenou robotickou úlohu - monitorování svého okolí a hledání objektů s potenciálně vysokou povrchovou teplotou.
- Možnost využití podvozku i pro jiné úlohy a jeho snadná konfigurace pro účely těchto úloh.
- Pozemní provedení podvozku a jeho malé rozměry pro použití výhradně v místnosti či specializované aréně.

Po průzkumu trhu a zvážení těchto faktorů, jsem se rozhodl postavit robot s využitím podvozku Zumo.

#### 8.1.1 Pásový podvozek Zumo Robot

Pásový podvozek Zumo mne zaujal především svou kompaktností a širokou oblastí použití. Výrobce podvozku, firma Pololu, na svých webových stránkách podvozek prezentuje jako velmi univerzální a tuto vlastnost ukazuje na úlohách, jako jsou např. sledování čáry, řešení bludiště, balancující robot či robot určený pro zápasy Mini Sumo. Na internetu se lze ale setkat i s aplikacemi, kdy je podvozek Zumo doplněn grafickou kamerou, kterou detekuje a sleduje objekt určité barvy či tvaru.

Vzhledem k pásovému provedení lze podvozek označit jako ne příliš vhodný pro přesnou odometrii a určování přesné polohy v prostoru z důvodu prokluzu pásů při manévrování. Pro účely autonomní jízdy v prostoru a vyhledávání podezřelého objektu však nemožnost určení přesné polohy příliš nevadí.

Podvozek od svého uvedení na trh prošel určitým vývojem. V současnosti se lze setkat se třemi variantami, z nichž každá poskytuje oproti svému předchůdci několik vylepšení. Lze se setkat s podvozkem Zumo Robot, Zumo Robot v1.2 a Zumo 32U4. Já jsem zvolil variantu Zumo Robot v1.2, protože novější 32U4 v době výběru ještě nebyla na trhu a navíc není kompatibilní se zamýšlenou řídicí jednotkou. Použitá varianta má oproti svému předchůdci integrovaný gyroskop, jenž jsem plánoval pro řízení robota využít.



Obr. 27) Podvozek Zumo Robot v1.2[11]

### 8.1.2 Parametry podvozku Zumo Robot v1.2

Základem podvozku je plastové šasi, do jehož útrobu se vkládají 4 tužkové baterie velikosti AA pro napájení robota. Dále jsou na podvozku namontovány mikro DC motory s převodovkou, kola a tzv. Zumo Shield, což je DPS obsahující sadu snímačů, napájecí obvody a konektory pro připojení vývojového kitu Arduino. Volitelně pak lze k podvozku připevnit malou DPS se světlocitlivými snímači pro aplikace sledování čáry a čelní kovovou radlici, tzv. Zumo Blade, pro zápasy Mini Sumo.

Výrobce poskytuje velkou variabilitu dodání podvozku, kdy lze zakoupit buď kompletní podvozek, nebo výše zmíněné jednotlivé prvky samostatně. To je výhodné a finančně efektivní v případě potřeby náhradních dílů nebo například v případě návrhu vlastní DPS obsahující mikroprocesor apod. Cena kompletního podvozku činí přibližně 2500 Kč.

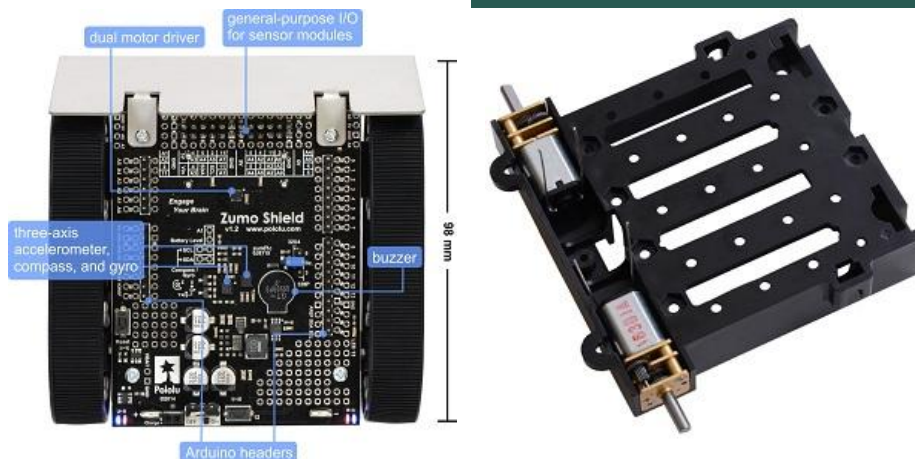
Tab 1) Mechanické parametry podvozku Zumo

Parametr	Hodnota
Rozměry	98 mm × 98 mm × 39 mm
Hmotnost	165 g

Zumo Shield obsahuje [11]:

- 3osý MEMS akcelerometr a kompas v jednom pouzdře – ST LSM303DLHC
- 3osý MEMS gyroskop – ST L3GD20H
- Duální H-můstek pro řízení dvou DC motorů – Texas Instruments DRV8835
- Dvě tlačítka, indikační LED, bzučák
- Napájecí obvody
- Konektory pro připojení světlocitlivých snímačů a pro připojení platformy Arduino





Obr. 28) Pohled na Zumo Shield a šasi s DC motory [11]

## 8.2 Mikro DC motory Pololu

Přímo pro zvolený podvozek výrobce dodává miniaturní DC motory, které je možné bez dalších úprav jednoduše připevnit k podvozku. Je možné vybrat si motor s oboustranně vyvedenou hřídelí nebo bez ní. Použité motory jsou ve variantě s oboustranně vyvedenou hřídelí, kde na jedné straně motoru je umístěna malá převodovka pro snížení jinak příliš vysokých mechanických otáček motoru a zvýšení krouticího momentu, a na druhé straně motoru je hřídel osazena rotačním enkodérem.

S ohledem na požadovanou rychlost robota a velikost krouticího momentu bylo třeba vybrat převodovku s vhodným převodovým poměrem. Zvolil jsem převodovku s poměrem 75,81 : 1. Výrobce u této převodovky stanovuje hodnotu otáček výstupní hřídele na 400 ot/min při napětí na motoru 6 V. Z toho lze určit napěťovou konstantu motoru  $K_V$ , jinými slovy přírůstek otáček na 1 V:

$$K_V = \frac{n}{U} = \frac{400}{6} = 66.66 \quad (17)$$

Kde  $K_V$  je napěťová konstanta  $[(\text{ot} \cdot \text{min}^{-1}) \cdot \text{V}^{-1}]$  motoru,  $n$  jsou otáčky motoru a  $U$  je napájecí napětí [V].

U navrženého robota je výkon motoru řízen PWM signálem o amplitudě 5V. Vezme-li se v úvahu napěťová konstanta motoru a 100% střídu PWM signálu, pak lze ze vztahu (17) vyjádřit otáčky a zjistit jejich maximální hodnotu při 5 V:

$$n_{5V} = K_V \cdot U = 66.66 \cdot 5 = 333.33 \text{ ot} \cdot \text{min}^{-1} = 5,56 \text{ ot} \cdot \text{s}^{-1} \quad (18)$$

Průměr kol, po kterých jsou vedeny pásy, činí 35 mm. Přičte-li se k této hodnotě ještě tloušťku pásů, je výsledná hodnota 39 mm. Tak lze vypočítat, jakou vzdálenost urazí odvalující se kolo za jednu otáčku, resp. celkový obvod kola:

$$O = \pi \cdot d = 3,14 \cdot 0,039 = 0,126 \quad (19)$$

Kde  $O$  je obvod kola [m] a  $d$  je průměr kola včetně pásu [m].

Ze znalosti obvodu kola a jeho maximálních otáček za jednotku času lze určit maximální rychlost, jakou se robot bude pohybovat v přímém směru bez prokluzu pásů:

$$v = O \cdot n_{5V} = 0,126 \cdot 5,56 = 0,70 \text{ m} \cdot \text{s}^{-1} \quad (20)$$

Kde  $v$  je maximální rychlost jakou je robot schopen vyvinout s použitým motorem a převodovkou [ $\text{m} \cdot \text{s}^{-1}$ ],  $O$  je obvod kola [m] a  $n_{5V}$  jsou otáčky motoru při 5 V [ $\text{ot} \cdot \text{s}^{-1}$ ].

Tab 2) Parametry použitých DC motorů při 6 V

Parametr	Hodnota
Jmenovité otáčky motoru za převodovkou	400 ot/min
Proudový odběr bez zátěže	120 mA
Maximální proudový odběr	1600 mA
Maximální krouticí moment	0,156 Nm
Převodový poměr převodovky	75,81:1
Průměr hřídele	1 mm
Průměr hřídele za převodovkou	3 mm

### 8.3 Magnetický enkodér Pololu

Navržený robot je osazen inkrementálními magnetickými rotačními enkodéry, které Firma Pololu nabízí ke svým motorům. Enkodéry se umísťují na volný konec oboustranně vyvedené hřídele motoru. Enkodér sestává z šestipólového magnetického kotoučku a malé destičky plošného spoje. Na destičce je umístěna dvojice Hallových senzorů obsahující napěťové komparátory s hysterezí, která zabraňuje generování zákmitů v případě, že se motor zastaví na hranici přechodu mezi opačně orientovanými póly magnetického kotoučku. Enkodér generuje během jedné otáčky magnetického kotoučku 3 přesné obdélníkové impulzy na každém z dvojice výstupních kanálů. To odpovídá 6 hranám či tzv. tikům na každém kanálu.



Obr. 29) Použitý DC motor a enkodér Pololu připevněný k motoru [10]

Použití enkodérů bylo podníceno potřebou měření rychlosti pro účely regulace pojezdu robota. Směr otáčení motorů je vždy vynucen řídicím SW a není ho tedy třeba sledovat. Použité enkodéry od počátku vyhovovaly jak po mechanické tak i po elektronické stránce.

Tab 3) Parametry použitých enkodérů

Parametr	Hodnota
Napájecí napětí	2.7V - 18V
Počet pólů mag. kotoučku	6
Počet kanálů	2
Počet tiků/ot	12
Rozměry enkodéru	12 mm x 10 mm

## 8.4 Řídící jednotka

Pro řízení robota jsem od začátku plánoval použít některý z mikrokontrolérů firmy Freescale (dnes již NXP), jelikož s mikrokontroléry tohoto výrobce mám mnoho praktických zkušeností z předchozích projektů. Navíc Freescale ke každé rodině MCU poskytuje široké portfolio vývojových desek. Tyto jsou především určeny pro použití v počátečních fázích vývoje projektů, kdy je zákazníkovi poskytnuta kvalitní prototypová základna, jež výrazně usnadňuje a urychluje prvotní vývoj. Není pak totiž nutné na začátku vývoje řešit návrh plošného spoje komplexní elektroniky a oživování samotného MCU. Tím je možné soustředit se od začátku vývoje přímo na požadovanou funkcionalitu cílového zařízení. Případnou další nutnou elektroniku lze vyvíjet formou samostatných a jednouchých modulů, které se k hlavní vývojové desce potom připojují prostřednictvím rozšiřujících konektorů. Tento přístup je také výhodný z ekonomického hlediska, protože samotná deska plošného spoje v praxi také podléhá vývoji a v naprosté většině případů se ji nepodaří na první pokus navrhnout a vyrobit v plně vyhovující podobě a funkcionalitě.

Z těchto důvodů jsem upřednostnil použití dostupného kitu, před vývojem vlastní desky plošných spojů. Navíc toto řešení sebou přineslo jednu důležitou výhodu. Použitý podvozek, resp. jeho elektronika, je původně určen k propojení s vývojovou deskou Arduino. Portfolio produktů firmy Freescale však obsahuje desku FRDM-KL25Z, která je v případě rozšiřujících konektorů a na nich vyvedených signálů plně kompatibilní s platformou Arduino. Proto je možné použít ji bez omezení pro řízení podvozkové desky Zumo Shield. Tato fakta umocnila moje rozhodnutí použít vývojový kit FRDM-KL25Z.

### 8.4.1 Vývojový kit FRDM-KL25Z

Vývojový kit FRDM-KL25Z představuje cenově dostupnou vývojovou platformu, kterou firma Freescale dodává pro „low-power“ mikrokontroléry z rodiny Kinetis L postavené na jádře ARM Cortex-M0+. Kit může být osazen některým 32bitovým MCU ze skupin KL14, KL15, KL24 nebo KL25, přičemž standardně je osazen nejvýkonnější možný s označením MKL25Z128VLK4. Technické parametry kitu čerpány z pramene [13]. Shrnutí základních parametrů použitého kitu:

- 32bitový MCU MKL25Z128VLK4, 48MHz, 128KB flash, 16KB SRAM, osmdesátipinové LQFP pouzdro
- Kapacitní dotykový slider, akcelerometr Freescale MMA8451Q, RGB LED
- Flexibilní možnosti napájení – USB, knoflíková baterie, externí zdroj
- Snadno přístupné MCU I/O piny prostřednictvím rozšiřujících konektorů
- Sofistikované OpenSDA debug rozhraní, umožňující ladění aplikace za běhu, ale i rychlé programování aplikace do paměti MCU v tzv. mass storage módu přes USB
- Rozměrová a pinová kompatibilita rozšiřujících konektorů s platformou Arduino
- Rozměry 82x54 mm



Obr. 30) Vývojová deska FRDM-KL25Z [13]

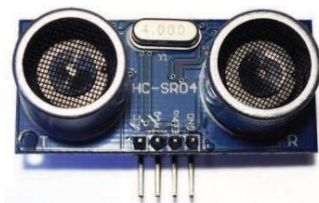
#### 8.4.2 Mikrokontrolér MKL25Z128VLK4

Tento mikrokontrolér patří do rodiny Kinetis L, konkrétně KL25, která je vhodná především pro použití v „low-power“ aplikacích nebo v aplikacích, které vyžadují komunikaci přes USB rozhraní. Stručný přehled základních parametrů mikrokontroléru [14]:

- 32bitové jádro ARM Cortex-M0+, pracovní frekvence až 48MHz
- Řadič přerušení ARM NVIC
- Interní 128kB flash a 16kB SRAM paměť
- Podpora několika režimů nízkého příkonu
- Vestavěný DMA řadič pro přímý přístup do paměti
- Vestavěný modul pro generování a distribuci časování pro jádro a ostatní periferie
- 16bitový SAR A/D převodník
- Analogový napěťový komparátor
- 12bitový D/A převodník
- Tři vícekanálové (1x6 a 2x2) moduly časovače/PWM
- RTC časovač
- Časovač s malým příkonem pro běh v low-power módech
- Dvoukanálový časovač pro generování periodického přerušení
- 2 x SPI
- 2 x I2C
- 3 x UART
- USB řadič včetně USB napěťového regulátoru
- Modul pro obsluhu kapacitních tlačítek

#### 8.5 Sonar HC-SR04

Autonomně řízený robot musí být schopen rozpoznat případné překážky a vyhýbat se jim. K detekci překážek je u navrhovaného robota použit ultrazvukový sonar HC-SR04. Sonar umožňuje měřit vzdálenost v rozsahu 2 cm až 4 m a detekuje objekt velikosti 3 x 3 cm na vzdálenost přibližně 2 m. Zorný úhel sonaru činí 15° [1].

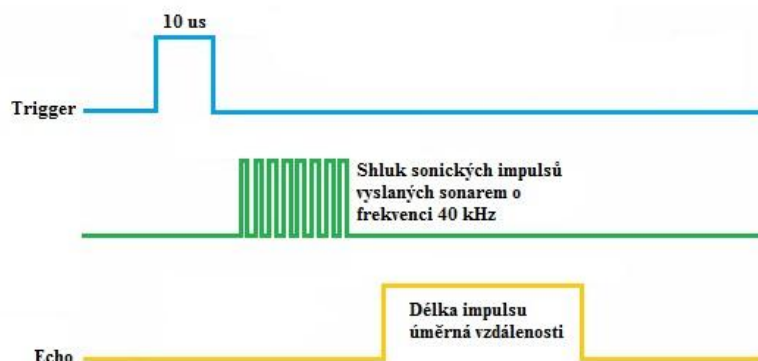


Obr. 31) Ultrazvukový sonar HC-SR04

### 8.5.1 Obsluha sonaru

Ovládání a činnost sonaru probíhá následujícím způsobem:

- Pro zahájení měření je nutné vygenerovat na vstupu Trigger pulz (log. 1) o délce minimálně 10  $\mu$ s.
- Po přijetí spouštěcí pulzu, vyšle sonar shluk osmi ultrazvukových impulzů směrem k překážce a čeká na jejich odraz.
- Jakmile sonar zachytí odražené ultrazvukové impulzy, vygeneruje pulz na výstupu Echo. Délka pulzu je úměrná vzdálenosti sonaru od překážky.

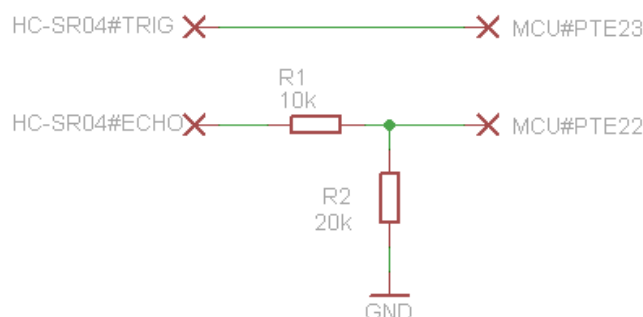


Obr. 32) Časová posloupnost signálů pro práci se sonarem

### 8.5.2 Převodník úrovní

Modul sonaru vyžaduje napájení 5 V a pracuje s úrovněmi TTL, zatímco použitý MCU pracuje s úrovněmi 3,3 V. Před připojením sonaru k MCU bylo nutné vyřešit tuto nekompatibilitu. V případě signálu Trigger a odpovídajícího pinu MCU nebyl nutný žádný zásah. Pin, ke kterému je signál připojen, je nastaven jako výstupní a sonar dokáže bezpečně zpracovat spouštěcí pulz s menší amplitudou než je 5 V.

Jiná situace je v případě signálu Echo. Odpovídající pin MCU je konfigurován jako vstupní a dlouhodobé působení vyššího napětí než 3,3 V by ho mohlo zničit. Pro ochranu tohoto pinu byl tedy navržen jednoduchý převodník úrovní v podobě napěťového děliče. Pro přetransformování napětí 5 V na 3,3 V, je třeba napětí na děliči rozložit v poměru 1/3 a 2/3. Dle této úvahy pak byly stanoveny hodnoty rezistorů  $R_1 = 10 \text{ k}\Omega$  a  $R_2 = 20 \text{ k}\Omega$ .



Obr. 33) Převodník napěťových úrovní pro připojení sonaru k MCU

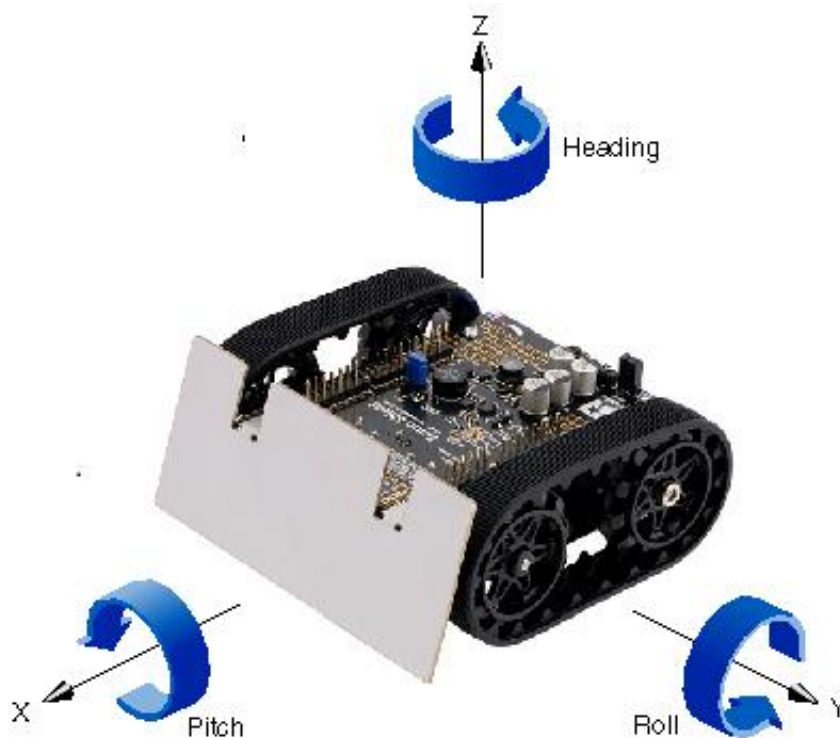
## 8.6 Gyroskop STMicroelectronics L3GD20H

Aby bylo možné u navrhovaného robota řídit směr jízdy, je nutné měřit údaj o jeho natočení. K měření jsem využil MEMS (z angl. Micro-Electro-Mechanical Systems) gyroskop, který je

osazen na podvozkové desce Zumo Shield. Jedná se o tříosý gyroskop s digitálním výstupem po sběrnici I2C a SPI. Základní parametry gyroskopu jsou [24]:

- Napájecí napětí 2,2 V – 3,6 V
- Volitelný měřicí rozsah  $\pm 245/\pm 500/\pm 2000$  DPS
- Volitelná vzorkovací frekvence 12,5 Hz – 800 Hz
- Konfigurovatelný vnitřní hornopropustný a dolnopropustný filtr
- Měřená data je možné ukládat do bufferu a odtud vyčítat více dat najednou
- Programovatelné přerušení
- Možnost převést gyroskop do low-power módu

K měření natočení byl použit pouze gyroskop. Využití akcelerometru pro tento účel nebylo možné, protože akcelerometr detekuje pouze změny náklonu kolem os rovnoběžných k zemskému povrchu - tedy náklony kolem os X a Y v Kartézském systému souřadnic. Kolmo na tyto osy působí gravitační zrychlení a v případě změny náklonu se úměrně mění hodnota gravitačního zrychlení v dané ose. Natočení kolem osy Z, kolem které se robot při pohybu bude otáčet, tak akcelerometr není schopen detekovat, protože vektor gravitačního zrychlení je s touto osou rovnoběžný. Pro lepší vysvětlení je souřadný systém robota zobrazen na obrázku níže.



Obr. 34) Souřadný systém robota



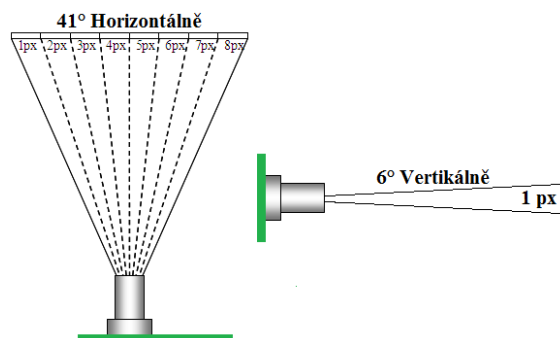
## 8.7 TPA81 Thermopile Array

Pro snímání okolní teploty bylo třeba robota vybavit vhodným bezdotykovým senzorem. Zvolil jsem senzor TPA81 Thermopile Array pro měření radiální teploty. Jedná se o komplexní modul, který obsahuje vlastní senzor a vestavěnou elektroniku s I2C rozhraním pro komunikaci s nadřazeným MCU.



Obr. 35) Thermopile Array TPA81[19]

Snímač detekuje infračervené záření v rozsahu vlnových délek  $2\text{ }\mu\text{m}$  až  $22\text{ }\mu\text{m}$ . Teplotní rozsah činí  $4\text{ }^{\circ}\text{C}$  až  $100\text{ }^{\circ}\text{C}$  s přesností  $\pm 2\text{ }^{\circ}\text{C}$ . Detektor je tvořen polem pixelů o rozměrech  $8 \times 1$ . Každý pixel obsahuje několik termočlánků zapojených do série. Zorný úhel každého pixelu je  $5,12^{\circ}$ , celkově pak  $41^{\circ}$  v horizontální rovině,  $6^{\circ}$  ve vertikální rovině. Výstupem je 8 hodnot radiální teploty z každého pixelu a jedna hodnota teploty okolního prostředí, kterou lze považovat za referenci. Modul je navíc vybaven výstupem pro ovládání serva, které lze využít pro otáčení modulu v 32 krocích kolem svislé osy a vytvářet tak teplotní mapu okolí. Tato praktická vlastnost podpořila výběr senzoru, neboť servo lze jednoduše řídit pomocí příslušného příkazu zaslaného po I2C sběrnici. Ušetří se tak 1 časovač a obsluha serva se celkově zjednodušuje. Čerpáno z [19].



Obr. 36) Zorné pole snímače [34]

### 8.7.1 Komunikace a obsluha

Jak bylo zmíněno výše, snímač lze ovládat prostřednictvím sběrnice I2C. Jedná se o sériovou synchronní sběrnici, která využívá dvou signálů – SDA (Synchronous Data) a SCL (Synchronous Clock). Standard I2C vyžaduje, aby oba signály byly přes pull-up rezistory připojeny k napájecímu napětí. Tím je zaručeno, že v době klidu na sběrnici budou oba signály v úrovni log. 1. Tato podmínka byla zajištěna externími rezistory o velikosti  $10\text{ k}\Omega$  na podvazkové desce robota.

Modul obsahuje sadu 10 registrů. Registry s adresou 0 a 1 jsou určeny jak pro čtení, tak pro zápis. Ostatní registry (adresa 2 až 9) slouží pouze pro čtení. Zápisem do registru 0 lze nastavit jednu z 32 pozic serva nebo pak speciální sekvenci zápisu změnit I2C adresu, která jednoznačně identifikuje senzor TPA81 na I2C sběrnici. Výchozí adresa je nastavena na hodnotu  $0xD0$ . Zápisem do registru 1 lze nastavit tzv. "Range" mód. V tomto režimu modul samostatně generuje obdélníkový signál pro otáčení servomotorem mezi požadovanými polohami a není nutné polohování obsluhovat. Tato vlastnost však nebyla využita, neboť bylo potřeba pozici serva neustále kontrolovat. Přehled registrů je uveden v následujících tabulkách:

Tab 4) Popis interních registrů senzoru TPA81

Register	Read	Write
0	Verze firmware	Pozice serva
1	Okolní teplota [°C]	Servo „Range“ mód
2	Pixel 1 [°C]	
3	Pixel 2 [°C]	
4	Pixel 3 [°C]	
5	Pixel 4 [°C]	
6	Pixel 5 [°C]	
7	Pixel 6 [°C]	
8	Pixel 7 [°C]	
9	Pixel 8 [°C]	

Pozice serva	Akce
0	Levá krajní poloha
n	Libovolná poloha (0; 31)
31	Pravá krajní poloha

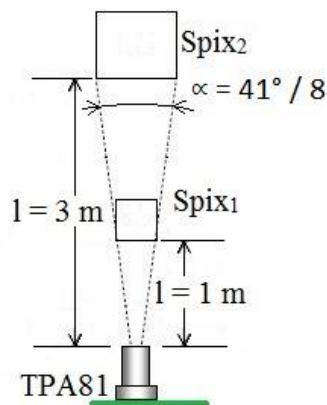
### 8.7.2 Vyhodnocení radiační teploty

Pro určení přesné povrchové teploty, musí robot přijet k měřenému objektu dostatečně blízko. K tomu aby rozhodl, jestli je dostatečně blízko či nikoliv, musel by robot znát plochu měřeného objektu z čelního pohledu a tuto porovnávat s plochou, kterou zabírá optika termočlánku.

Pro jednoduchost se uvažuje, že plocha snímaná 1 pixelem TPA81 je čtvercového tvaru<sup>2</sup>. Snímanou plochu pak lze vyjádřit [34]:

$$S_{pix} = l * tg(\alpha) \quad (21)$$

Kde  $S_{pix}$  je snímaná plocha jedním pixelem [m<sup>2</sup>],  $l$  je vzdálenost [m] a  $\alpha$  je zorný úhel [°] (v případě zjednodušeného snímače TPA81  $\alpha = 41^\circ / 8$ )



Obr. 37) Změna snímané plochy v závislosti na vzdálenosti [34]

Tento princip však nelze uplatnit pro obecné úlohy, kde nejsou známy rozměry měřeného předmětu. Pro přesné měření radiační teploty je třeba navíc znát emisivitu povrchu měřeného objektu, tu však v obecných podmínkách také nelze určit.

<sup>2</sup> Ve skutečnosti není, protože zorné pole je v horizontální i vertikální rovině různě velké viz popis optiky teploměru TPA81 v kapitole 8.7).



## 8.8 Bluetooth modul HC-05

Pro bezdrátovou komunikaci byl robot vybaven Bluetooth modulem HC-05. Modul lze provozovat jak v režimu slave, tak i v režimu master. V režimu slave může modul pouze přijmout žádost o spárování, kdežto v režimu master může modul žádost o spárování s dalším zařízením sám iniciovat. Ke komunikaci s MCU modul využívá sériovou linku UART. Ve výchozím stavu je sériová komunikace nastavena následovně:

- Rychlost 9600 baud
- 8 datových bitů
- 1 start bit
- Žádná parita

Výchozí rychlost však pro náročnější aplikace může být nevyhovující a je nutné ji zvýšit. Zvýšení rychlosti a další nastavení je možné provést v konfiguračním režimu prostřednictvím tzv. AT příkazů. Uvedení modulu do konfiguračního režimu docílí stiskem a podržením jediného dostupného tlačítka na modulu a následným připojením modulu k napájecímu napětí. V konfiguračním režimu modul neposkytuje Bluetooth konektivitu, ale pouze přijímá AT příkazy po sériovém rozhraní. Ukázka AT příkazu pro změnu rychlosti sériového rozhraní na rychlost 115200 baud:

```
AT+UART=115200
```

```
OK
```

Po každém validním přijatém AT příkazu modul odpoví zprávou OK a související nastavení je uloženo do interní nevolatilní flash paměti. Obsah této paměti bude uchován i po odpojení napájení a je tedy možné provést konfiguraci pouze jednou. Pro ukončení konfiguračního režimu je nutné provést reset modulu odpojením od napájecího napětí.

Tab 5) Parametry Bluetooth modulu HC-05

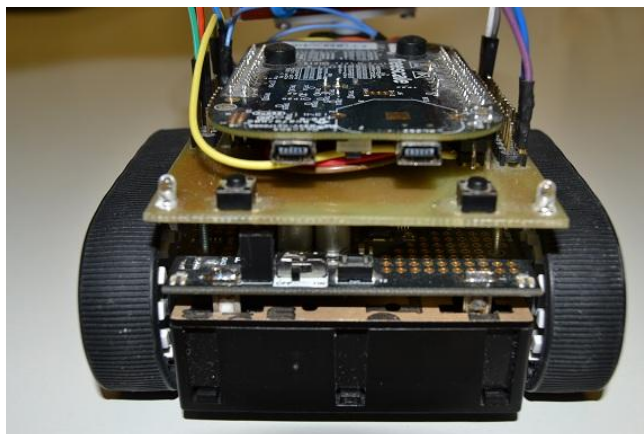
Parametr	Hodnota
Napájecí napětí	3.3 V – 6 V
Frekvenční pásmo	2.4 GHz
Maximální přenosová rychlost	3 Mbps
Maximální dosah	10 m
Výchozí rychlost sériové komunikace	9600 baud
Maximální rychlost sériové komunikace	1382400 baud
Rozměry modulu	12,7 mm x 27 mm



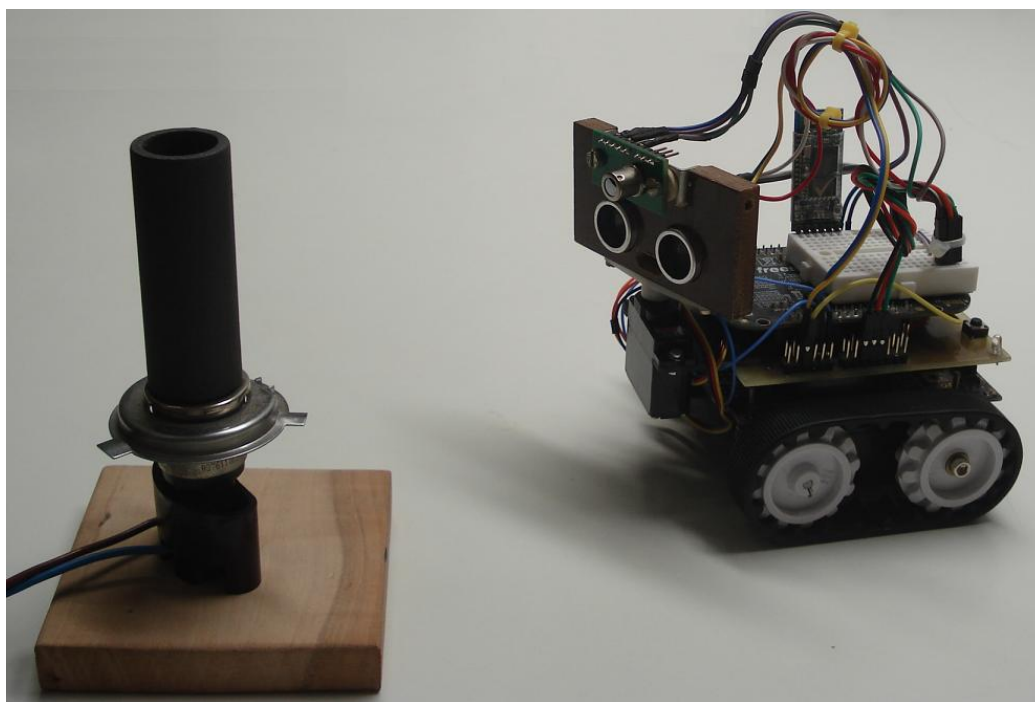
Obr. 38) Bluetooth modul HC-05 [35]

### 8.9 Rozšiřující deska pro Zumo Shield a FRDM-KL25Z

Při návrhu robota byla navržena a vyrobena velice jednoduchá rozšiřující deska plošných spojů s vyvedením všech signálů dostupných na konektorech řídicí desky FRDM-KL25Z. Tato deska tvoří mezivrstvu mezi řídicí a podvozkovou deskou. Důvodem pro výrobu této desky byla sendvičová konstrukce obou hlavních desek, které jsou vodivě propojeny průchozími konektory. Tato konstrukce znemožňovala bezpečné a trvalé připojení sond osciloskopu ke konektorům při ladění robota. Současně tato deska rozšiřuje napájecí konektory, jejichž počet nedostačoval pro připojení všech externích senzorů. Na desce jsou rovněž osazeny dvě indikační LED diody a dvě mikrotlačítka pro ovládání robota.



Obr. 39) Sendvičová konstrukce robota. Nahoře řídicí deska FRDM-KL25Z, uprostřed rozšiřující deska a dole podvozková deska Zumo Shield



Obr. 40) Fotografie navrženého robota a testovacího přípravku pro simulaci předmětu s vysokou povrchovou teplotou (žárovka H5)

## 9 REALIZACE ŘÍDÍCIHO FIRMWARE

### 9.1 Task scheduler

Řízení robota bylo rozděleno do několika úloh – úloh. Na základě zkušeností z jiných projektů byl pak implementován jednoduchý task scheduler (plánovač), který zajišťuje spouštění úloh podle určitého časového rozvrhu. Každý úkol je reprezentován následující datovou strukturou:

```
typedef struct task
{
    void          (*taskFunc)(void *); /* Task function */
    TaskStatus_t  taskStatus;          /* Task status */
    uint16_t      taskPeriod;          /* Default period */
    uint16_t      taskCnt;             /* Estimation time counter */
} TTASK;
```

Struktura obsahuje:

- ukazatel na funkci úlohy
- příznak, kterým lze provádění úlohy povolit či zakázat
- výchozí periodu, se kterou se bude provádět spouštění úlohy [ms]
- čítač zbývajících času před spuštěním úlohy [ms]

Následující fragment kódu zachycuje statickou inicializaci plánovače, tzn. jednotlivé úlohy, jejich výchozí stav, periodu a pracovní čítač. Význam jednotlivých úloh bude vysvětlen dále v textu v souvisejících kapitolách.

```
/* List of the tasks */
static TTASK taskList[] = {
    /* Task Function          Status    Period  Counter */
    { &CONTROL_ZumoControl,   kTaskEnabled,  10,    100 },
    { &L3GD20_MeasureHeading, kTaskEnabled,  10,     0 },
    { &TPA81_MeasureAmbientTemp, kTaskEnabled, 2000,  0 },
    { &TPA81_MeasureRadiationTemp, kTaskEnabled,  10,     0 },
    { &PACKET_HandlePacketIfReady, kTaskEnabled,  20,    100 },
};
```

Časování plánovače je odvozeno od časovače PIT0, který je nastaven tak, aby se každou milisekundu vyvolalo přerušení. V obsluze přerušení se všem povoleným úlohám sníží hodnota čítače zbývajících času `taskCnt`. V okamžiku, kdy je tento čítač vynulován, je příslušný úkol vložen na konec fronty `readyTasksQueue`. Tato fronta je implementována jako kruhový buffer a obsahuje všechny úlohy připravené ke spuštění.

Modul task scheduleru obsahuje hlavní programovou smyčku, ve které se testuje obsah fronty `readyTasksQueue`. Úlohy jsou pak z jejího počátku odebírány a spouštěny až do okamžiku úplného vyprázdnění fronty. Po dokončení úlohy je čítač zbývajících času `taskCnt` inicializován na hodnotu výchozí periody, která je uložena v položce `taskPeriod` a celý proces se opakuje.

```
void TASK_DispatchTasks(void)
{
    /* Init task queue */
    taskQueueHead = 0U;
    taskQueueTail  = 0U;

    /* Main program loop */
    while (1) {
        while (taskQueueHead != taskQueueTail) {
            /* Task execution */
            (*taskList[readyTasksQueue[taskQueueHead]].taskFunc) (NULL);

            /* Move on the next task */
            taskQueueHead++;

            if (taskQueueHead >= TASK_QUEUE_SIZE) {
                taskQueueHead = 0U;
            }
        }
    }
}
```

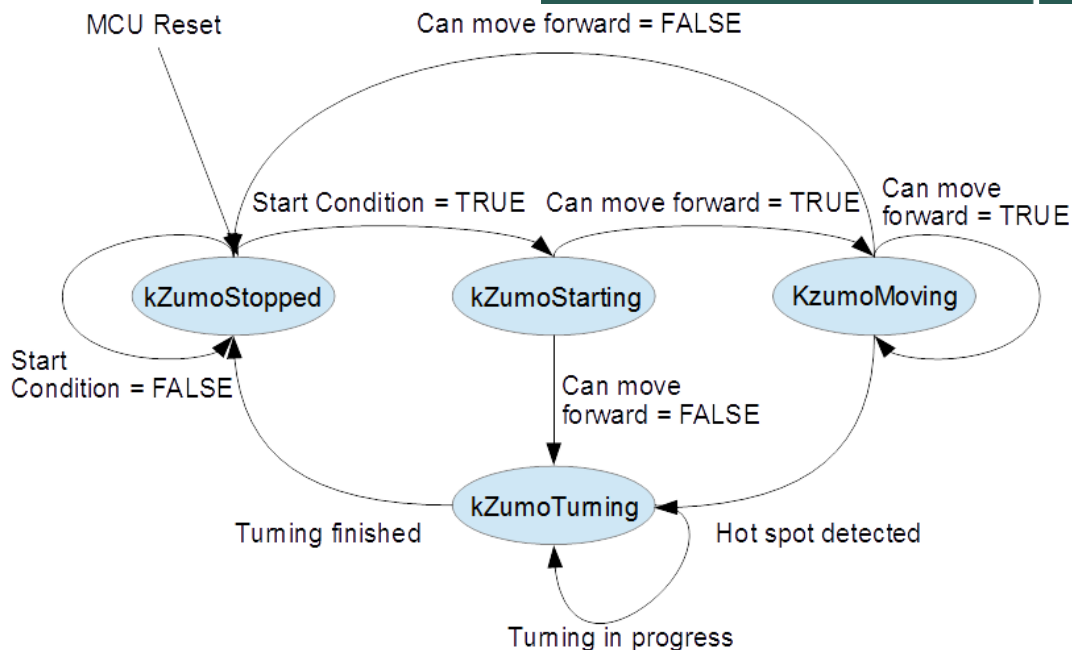
Modul plánovače implementuje také funkce pro povolení či zakázání určitého úkolu nebo funkci pro modifikaci čítače zbývajících času. Prostřednictvím těchto funkcí lze za běhu zasahovat do výchozího časového rozvrhu. Priority úkolů nejsou řešeny a platí že, nastane-li situace, kdy v jenom časovém okamžiku má být spuštěno více úkolů, tak budou tyto provedeny v pořadí, jak jsou definovány v poli `taskList` viz statická inicializace výše.

Tab 6) Stručný význam definovaných úkolů

<b>Úkol</b>	<b>Perioda [ms]</b>	<b>Význam</b>
<code>CONTROL_ZumoControl()</code>	10	Řízení robota, regulace
<code>L3GD20_MeasureHeading()</code>	10	Měření polohy (natočení podvozku)
<code>TPA81_MeasureAmbientTemp()</code>	2000	Měření okolní teploty
<code>TPA81_MeasureRadiationTemp()</code>	10	Polohování serva, měření radiační teploty, měření vzdálenosti od překážky
<code>PACKET_HandlePacketIfReady()</code>	20	Kontrola přijatých paketů a jejich zpracování

## 9.2 Řízení robota

Řízení je prováděno úkolem `CONTROL_ZumoControl()` implementovaným v modulu `control.c`. Po spuštění úkolu se vždy nejprve otestuje stav robota, přičemž robot se může nacházet v jednom ze čtyř stavů.



Obr. 41) Stavový diagram robota

Po resetu MCU a při prvním spuštění řídicího úkolu se robot nachází ve stavu `kZumoStopped`. V tomto stavu robot setrvává až do okamžiku splnění startovací podmínky. Ta může být splněna stiskem levého tlačítka na rozšiřující desce nebo přijetím start paketu sériovým rozhraním UART0. Po uplynutí jedné z možných událostí se nastaví řídicí proměnná na hodnotu `TRUE` a stav robota při dalším spuštění úkolu přejde do stavu `kZumoStarting`.

### 9.2.1 Rozjezd robota

Ve stavu `kZumoStarting` se nejprve testuje, zdali před robotem není žádná překážka a jestli se může rozjet. Tento test je prováděn funkcí `SONAR_RobotCanMoveForward()`, která porovnává poslední změřenou vzdálenost od možné překážky s prahovou hodnotou, která je ve výchozím stavu nastavena na 15 cm. Jestliže ve stanovené vzdálenosti není detekována žádná překážka, funkce vrací hodnotu `true`, jinak funkce vrací hodnotu `false`.

V případě, že žádná překážka před robotem detekována není, bude provedena inicializační sekvence rozjezdu robota. Ta spočívá v konfiguraci H-můstku do režimu dopředného chodu obou motorů nastavením příslušných řídicích signálů do log. 0. Dále bude vypočítán přírůstek otáčkové rampy pro plynulý rozjezd robota. Následuje nastavení proměnné reprezentující relativní úhel směru jízdy vzhledem k výchozí pozici na hodnotu  $0^\circ$  a inicializace všech regulátorů za účelem vynulování integračních a derivačních složek. Posledním krokem je vynulování počítadla neúspěšných pokusů o rozjezd. Po provedení této sekvence přejde řízení robota do stavu `kZumoMoving`.

Pokud funkce pro detekci překážky vrátí řídicímu úkolu hodnotu `false`, bude zvýšeno počítadlo neúspěšných pokusů o rozjezd. V případě, že počítadlo ve stavu `kZumoStarting` dosáhne hodnoty 5, provede robot otočení do jiného směru. Výpočet nového směru je proveden funkcí `CONTROL_CalcRandHeading()` pomocí jednoduchého generátoru pseudonáhodných čísel. Po výpočtu požadované hodnoty směrového úhlu následuje inicializace regulátorů a robot bude převeden do stavu `kZumoTurning`, ve kterém bude provedeno samotné otočení robota.

### Výpočet směrového úhlu pomocí generátoru pseudonáhodných čísel

Pro generování pseudonáhodných čísel byl převzat algoritmus [31] vycházející z metody lineárního kongruentního generátoru. Jedinou podmínkou tohoto algoritmu je existence 32bitového datového typu integer. Tato podmínka je na zvolené MCU platformě a s použitým GCC kompilátorem splněna.

```
static int CONTROL_RandomInt32(void)
{
    unsigned int hi, lo;
    static int rndSeed = 1; /* Initial seed of pseudo random generator*/

    hi = 16807 * (rndSeed >> 16);
    lo = 16807 * (rndSeed & 0xFFFF);
    lo += (hi & 0x7FFF) << 16;
    lo += hi >> 15;
    if (lo > 2147483647) {
        lo -= 2147483647;
    }
    rndSeed = lo;

    return rndSeed;
}
```

Vygenerované pseudonáhodné číslo slouží jako žádaná hodnota směrového úhlu. Před jeho použitím ho však bylo třeba převést do vhodného intervalu  $\langle -270^\circ; -30^\circ \rangle \cup \langle 30^\circ; 270^\circ \rangle$ . Záporná část intervalu představuje otočení robota doleva od stávající pozice, kladná část pak odpovídá otočení doprava. Hodnoty v intervalu  $(-30^\circ; 30^\circ)$  jsou zanedbány, jelikož otočení o tak malý úhel v naprosté většině nevedlo k vyhnutí se překážce. Navíc bylo velmi problematické regulovat otočení pásového podvozku o tak malý úhel. Hodnoty menší než  $-270^\circ$ , resp. větší než  $270^\circ$  byly rovněž saturovány. Zamezilo se tak situaci, kdy se robot mohl otočit dokola a zastavit se ve stejné pozici jako před otočením. Rovněž otočení o více jak  $360^\circ$  je zbytečné.

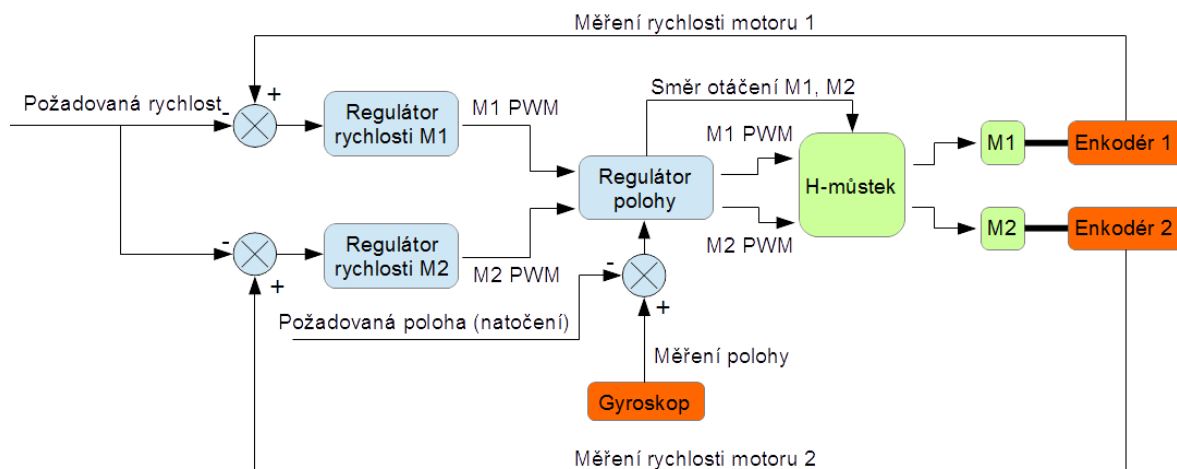
```
...
val = CONTROL_RandomInt32() % 540;
val -= 270;

if((val > -30.0) && (val <= 0.0)) val = -30.0;
if((val > 0.0) && (val < 30.0)) val = 30.0;
```

### **9.2.2 Řízení jízdy – regulace rychlosti a polohy**

Řízení jízdy je řešeno ve stavu `kZumoMoving`. V tomto stavu se nejprve vyhodnocuje přítomnost překážky před robotem. Jestliže je detekována překážka, robot bude okamžitě zastaven a převeden do stavu `kZumoStarting`, ve kterém proběhne výpočet nového směru

jízdy. Pokud se v dovolené vzdálenosti před robotem žádná překážka nevyskytuje, tak se provede regulace rychlosti a polohy robota.



Obr. 42) Blokové schéma regulace jízdy robota

### Rychlostní regulátory

Pro regulaci rychlosti byla implementována dvojice diskrétních PSD regulátorů. Každý motor má tak samostatný regulátor. Společným vstupem obou regulátorů je žádaná rychlost robota. Pro plynulý rozjezd robota je žádaná rychlost z počátku pozvolna zvyšována až do okamžiku dosažení nominální rychlosti a nadále až do zastavení robota zůstává konstantní. Druhým vstupem každého z dvojice regulátorů je aktuální rychlost příslušného motoru. Metoda měření rychlosti je popsána v kapitole 9.3. Ladění regulátorů bylo provedeno metodou Ziegler-Nichols.

Prvním krokem rychlostního regulátoru je výpočet regulační odchylky. Následuje výpočet proporcionální, integrační a derivační složky. Výpočet integrační složky je prováděn jednoduchou obdélníkovou metodou (sumace všech hodnot regulační odchylky od začátku regulace). Součtem všech tří složek regulátoru je získán regulační zásah v podobě střídý PWM. Hodnota střídý PWM je však ještě před nastavením do periferie PWM modulátoru nejprve korigována polohovým regulátorem. Součástí regulátoru je i Anti-Windup opatření, které kontroluje stav regulátoru. Pokud regulační zásah přesáhne dovolený limit střídý PWM (0 až 95%), dojde vlivem Anti-Windup opatření k dočasnému deaktivování integrace regulační odchylky. Integrace bude opět povolena v okamžiku, kdy se velikost akční veličiny (střída PWM) vrátí do normálních mezí. Pro generování PWM pro DC motory byla využita periferie TPM0 (Timer/PWM Module). Frekvence PWM signálů byla stanovena na 20 kHz.



```

uint16_t PIDREG_M1RpmController(uint32_t actualSpeed, uint32_t setpoint)
{
    int32_t err, p, i, d, pwm;

    err = (int32_t) actualSpeed - (int32_t) setpoint; /* Actual error */

    /* P */
    p = (int32_t) (M1RpmPid.pGain * (float) err);

    /* D */
    d = M1RpmPid.dGain * (err - M1RpmPid.lastErr);
    M1RpmPid.lastErr = err;

    /* Anti-Windup */
    if ((M1RpmPid.state == kPidUpperLimit) && (err > 0)) {
        err = 0;
    }
    if ((M1RpmPid.state == kPidLowerLimit) && (err < 0)) {
        err = 0;
    }

    /* I */
    M1RpmPid.sumErr = M1RpmPid.sumErr + err;
    i = M1RpmPid.iGain * M1RpmPid.sumErr;

    pwm = p + i + d; /* New output value of the controller */

    /* Set default controller state */
    M1RpmPid.state = kPidNormalRange;

    /* Check whether new output value is in normal range, if not, change
    controller state */
    if (pwm > (int32_t) TPM_MAX_PWM) {
        M1RpmPid.state = kPidUpperLimit;
        pwm = (int32_t) TPM_MAX_PWM;
    }
    if (pwm < 0) {
        M1RpmPid.state = kPidLowerLimit;
        pwm = 0;
    }
    return ((uint16_t) pwm);
}

```



### Polohový regulátor

Polohový regulátor slouží pro řízení směru jízdy robota. Vstupem je žádaná a aktuální hodnota směrového úhlu změřená gyroskopem viz kapitola 9.4. Dále do regulátoru vstupují hodnoty PWM vypočtené rychlostními regulátory. Polohový regulátor provede korekci těchto hodnot dle požadovaného směru jízdy a až poté jsou vypočtené střídly PWM nastaveny do periferie PWM modulátoru. Žádaná hodnota směru jízdy je ovlivňována stavem detekce podezřelého objektu. Před provedením regulačního zásahu se vždy nejprve zjistí stav detekce a případně se modifikuje žádaná hodnota směru jízdy. Proces korekce směru jízdy je popsán v kapitolách 9.6.4 a 9.6.5.

Z implementačního hlediska se jedná o obdobu rychlostního regulátoru. Ladění regulátoru bylo opět provedeno metodou Ziegler-Nichols s manuální doladěním. Výstupem regulátoru jsou hodnoty stříd PWM. Navíc regulátor poskytuje informaci o požadovaném smyslu otáčení pro H-můstek. Tato informace je nutná v okamžiku, kdy se provádí polohovým regulátorem otáčení robota na místě a je v takovém případě nutné, aby chod obou motorů byl vzájemně opačný. Určení směru chodu motorů ukazuje následující úsek kódu polohového regulátoru.

```
void PID_HeadingController(float actHeading, float reqHeading, uint16_t
*reqM1Pwm, uint16_t *reqM2Pwm)
{
    ...
    /* Manipulated variable */
    u = p + i + d;

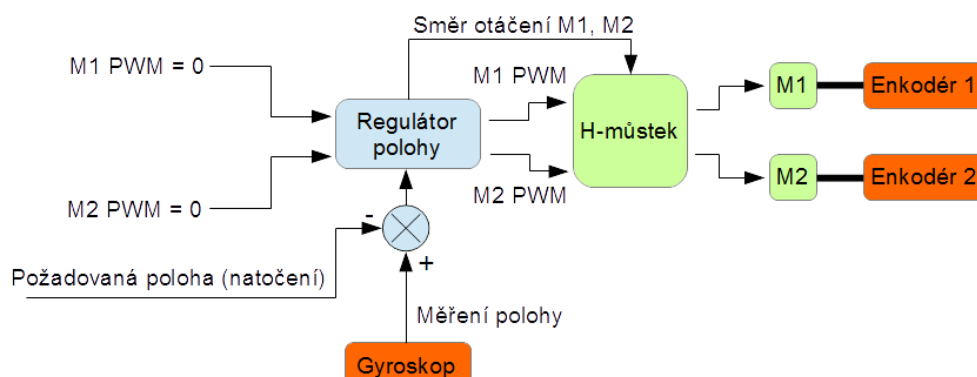
    m1Pwm = (int32_t) *reqM1Pwm + u;
    m2Pwm = (int32_t) *reqM2Pwm - u;

    if (m1Pwm < 0) {
        GPIO_SET_M1DIR_BACKWARD();
        m1Pwm *= (-1);
    } else {
        GPIO_SET_M1DIR_FORWARD();
    }
    if (m2Pwm < 0) {
        GPIO_SET_M2DIR_BACKWARD();
        m2Pwm *= (-1);
    } else {
        GPIO_SET_M2DIR_FORWARD();
    }

    *reqM1Pwm = (uint16_t) m1Pwm;
    *reqM2Pwm = (uint16_t) m2Pwm;
}
```

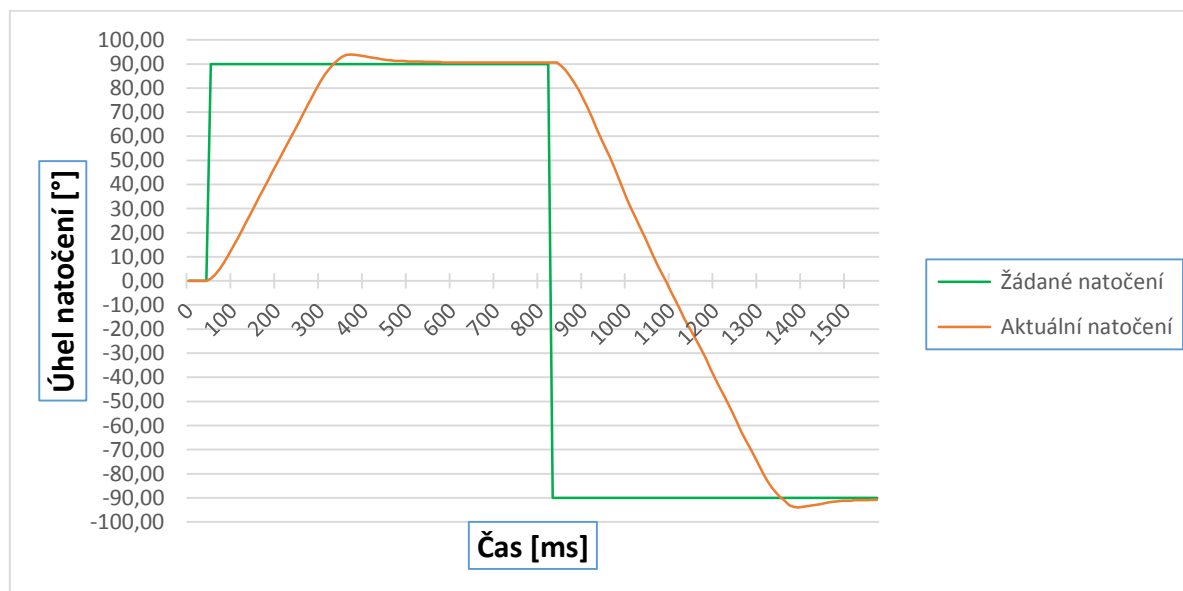
### 9.2.3 Otáčení robota

Při požadavku na otočení podvozku na místě je robot převeden do stavu `kZumoTurning`. V tomto stavu probíhá pouze regulace polohy, regulace rychlosti je vyřazena z činnosti. Před spuštěním směrového regulátoru se vždy nejprve vyčtou data z gyroskopu a zjistí se aktuální natočení podvozku. Hodnota aktuálního směrového úhlu se porovná s požadovanou hodnotou. Pokud není robot natočen do požadované polohy, bude spuštěn regulátor polohy. Výstupem regulátoru je nová střída PWM signálů pro oba motory včetně informace o požadovaném smyslu otáčení pro H-můstek.



Obr. 43) Blokové schéma regulace otáčení robota

Regulace končí v okamžiku, kdy začne úspěšně procházet test na rovnost aktuálního a požadovaného natočení s dovolenou tolerancí  $\pm 2^\circ$ . Pokud podmínce vyhoví 20 po sobě jdoucích měření, je natočení robota považováno za dokončené. Robot pak přejde do stavu `kZumoStopped`, resp. do `kZumoStarting` odkud se robot pokusí znovu rozjet.



Obr. 44) Průběh regulace při skokové změně žádané polohy (natočení)

### 9.3 Měření rychlosti

Rychlost každého z obou motorů je měřena nezávisle pomocí dvojice inkrementálních enkodérů. Pro měření rychlosti postačuje sledovat pouze jeden výstupní kanál a je třeba na něm měřit dobu mezi dvěma změnami obdélníkového signálu nebo obdélníkový signál

vzorkovat a vyhodnocovat počet impulzů či hran za jednotku času. Rozhodl jsem se pro první zmíněnou strategii.

K měření je využíván dvoukanálový časovač TPM1, nastavený do tzv. Input-Capture režimu. Oba kanály využívají společný 16bitový čítač, který je nastaven jako volně běžící a inkrementuje se s každým tikem referenčního hodinového signálu o frekvenci 6 MHz. S každým kanálem je asociován jeden pin, na který je přiveden výstupní signál z příslušného enkodéru. Jedná se o piny PTE20 a PTE21.

Při každé náběžné hraně signálu z enkodéru se vyvolá přerušení MCU od souvisejícího kanálu. V obsluze přerušení se vždy vyčte aktuální hodnota čítače. Současně je potřeba uchovávat i hodnotu, jakou čítač obsahoval při obsluze předcházející události. Z rozdílu těchto dvou hodnot lze pak zjistit, kolik tiků čítač načítal během dvou po sobě jdoucích hranách obdélníkového signálu. Velikost otáček je tedy reprezentována jako počet tiků čítače během dvou hran kvadrurního signálu a platí, že čím vyšší otáčky, tím menší je počet tiků.

Protože je čítač konfigurován jako volně běžící a přerušení jsou vyvolávána asynchronně, je potřeba do výpočtu rychlosti zahrnout i případná přetečení čítače, která mohou nastat mezi dvěma událostmi. Při každém přetečení čítače se tedy opět vyvolá přerušení, v jehož obsluze se inkrementuje počítadlo přetečení a toto počítadlo dále figuruje ve výpočtu skutečného počtu tiků. Tento výpočet probíhá v obsluze přerušení časovače TPM1 a pro jeden enkodér je implementován následujícím fragmentem kódu:

```
/* Store previous counter value */
ElTicksPrev = ElTicksAct;
/* Get current value of the counter */
ElTicksAct = TPM1_C1V;

if (ElTicksPrev > ElTicksAct) {
    ElTicksDelta = (uint32_t) 65536 - ElTicksPrev + ElTicksAct;

    if (ElTimerOverflowCnt > 0U) {
        ElTimerOverflowCnt--;
    }
} else {
    ElTicksDelta = ElTicksAct - ElTicksPrev;
}

/* Calculate time between two rising edges including timer overflow */
ElTicksDelta = (ElTimerOverflowCnt * (uint32_t) 65536) + ElTicksDelta;
/* Update circular speed buffer */
TPM_UpdateElSpeedBuf(ElTicksDelta);
/* Reset overflow counter */
ElTimerOverflowCnt = 0;
```

Vypočtená hodnota rychlosti se ukládá pomocí funkce `TPM_UpdateExSpeedBuf()` do kruhového bufferu a z hodnot v bufferu se poté počítá aritmetický průměr pro filtraci měřených hodnot. Experimentálně byl vyzkoušen i mediánový filtr pro eliminaci extrémních hodnot, ale na měření a následné regulaci rychlosti se tato změna nijak výrazně neprojevila.

```
/**
 * Function stores last value of Enc 1 speed into circular buffer.
 */
static void TPM_UpdateE1SpeedBuf(uint32_t speed)
{
    E1SpeedBuf[E1SpeedBufIdx] = speed;
    E1LastData = E1SpeedBufIdx;
    E1SpeedBufIdx++;

    if (E1SpeedBufIdx >= TPM1_SPEED_BUF_SIZE) {
        E1SpeedBufIdx = 0;
    }
}

/**
 * Function returns average speed of Enc(motor) 1. Returned value is
 * calculated as average of last 8 measured values. During RPM speed ramp
 * will be returned last measured value, not average value.
 */
uint32_t TPM_GetE1Speed(void)
{
    uint32_t retVal = 0U;

    NVIC_DisableIRQ(TPM1_IRQn); /* Disable interrupt from TPM1 */
    if (rampEnd) {
        /* Get sum of values in buffer */
        for (uint8_t idx = 0; idx < TPM1_SPEED_BUF_SIZE; idx++) {
            retVal += E1SpeedBuf[idx];
        }
        /* Sum of values divided by TPM1_SPEED_BUF_SIZE */
        retVal = retVal / TPM1_SPEED_BUF_SIZE;
    } else {
        retVal = E1SpeedBuf[E1LastData];
    }
    NVIC_EnableIRQ(TPM1_IRQn); /* Enable interrupts */
    return retVal;
}
```

## 9.4 Měření natočení a obsluha gyroskopu

K měření natočení podvozku byl využit MEMS gyroskop popsáný v kapitole 8.6. Před vlastním měřením bylo nutné gyroskop inicializovat.

### 9.4.1 Inicializace gyroskopu

Prvním krokem při inicializaci gyroskopu byl výběr rozhraní, po kterém bude MCU s gyroskopem komunikovat. Na výběr byla sběrnice SPI či I2C. Konkrétní varianta se řídí napětíovou úrovní na pinu SC. Výrobce desky Zumo Shield připojil tento pin přes externí pull-up rezistor k napájecímu napětí 3.3 V a bylo proto zvoleno rozhraní I2C - viz schéma [12] a referenční manuál [24].

Inicializace prakticky spočívala v nastavení měřicího rozsahu, nastavení vzorkovací frekvence a nastavení vestavěných filtrů. Proběhla komunikací po zvoleném rozhraní a zápisem do vnitřních registrů gyroskopu.

Měřicí rozsah byl stanoven na  $\pm 500$  dps. Nižší měřicí rozsah  $\pm 245$  dps, který poskytuje vyšší rozlišení, nemohl být použit, neboť bylo zjištěno, že otáčení robota při plném výkonu motorů je příliš rychlé a tento měřicí rozsah nepostačuje.

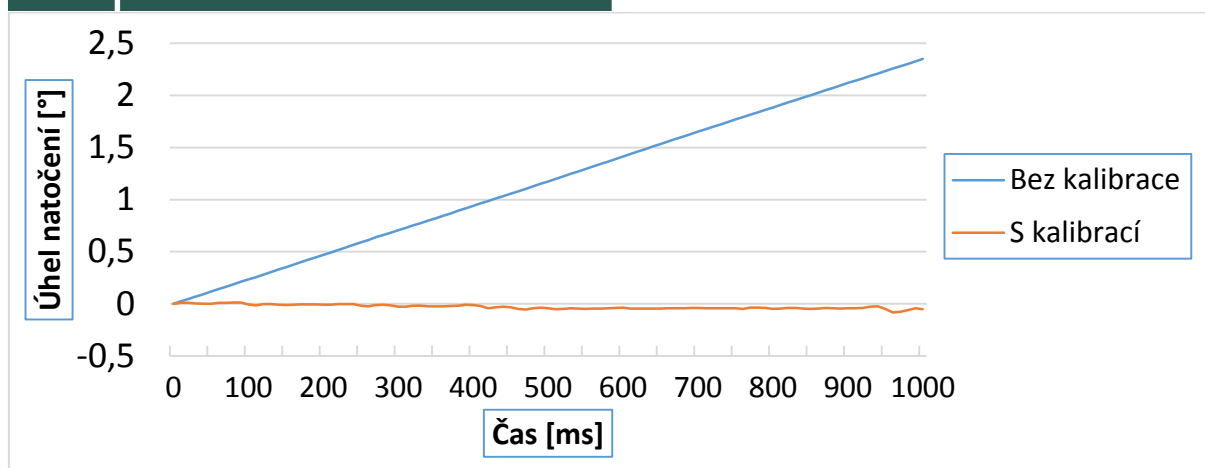
Vzorkovací frekvence, tedy rychlost měření, byla stanovena na 100Hz. To odpovídá periodě 10ms. Se stejnou periodou se provádí i úkoly pro vyčtení naměřených hodnot po I2C a regulační zásah směrového regulátoru. Je tak zajištěno, že ve výpočtech budou použita vždy pouze aktuální data.

Vestavěné konfigurovatelné hornopropustní a dolnopropustní filtry byly po několika pokusech o jejich co nejoptimálnější nastavení zcela vypnuty, neboť jejich použití nevedlo ke zvýšení přesnosti měření, ale právě naopak. Při jejich použití gyroskop ignoroval pomalé nebo příliš rychlé změny natočení. Tyto filtry je výhodné použít v případě, kdy se k měření úhlového zrychlení používá více senzorů. Jeden senzor pak může ořezat všechny vysokofrekvenční změny, druhý senzor naopak nízkofrekvenční změny a jsou-li pak tato data vhodně zkombinována např. Kalmanovým filtrem, je možné dosáhnout přesnějšího měření než v případě použití jediného gyroskopu.

Gyroskop dále umožňuje vyvolávat externí přerušení při určitých událostech nebo ukládat data do FIFO front apod. Tyto vlastnosti však nebyly využity.

### 9.4.2 Kalibrace gyroskopu

Po inicializaci bylo nutné provést kalibraci gyroskopu kvůli potlačení driftu. Kalibrace je řešena tak, že se z gyroskopu vyčte 100 surových hodnot úhlové rychlosti. Z těchto hodnot je následně vypočítán aritmetický průměr a výsledná hodnota je považována za offset, který se vždy při již precizním měření dle znaménka přičte nebo odečte od měřených dat. Vliv kalibrace na gyroskop, který setrvává v klidu, je znázorněn na obrázku níže.



Obr. 45) Vliv kalibrace na gyroskop a potlačení driftu

### 9.4.3 Výpočet úhlu natočení

Výpočet úhlu natočení je prováděn integrací úhlové rychlosti v čase viz kapitola 6.2.1. Vyčtení měřených dat a jejich přepočítání obstarává úkol `L3GD20_MeasureHeading()`. Gyroskop měří ve všech třech osách, ale z důvodu snížení režie, která je spojená s komunikací po I2C, jsou z gyroskopu vyčítána data pouze pro osu Z. Nicméně, kód je připraven tak, že jednoduchým způsobem lze povolit čtení i zbývajících os X a Y. Po přečtení dat se provede korekce podle kalibrace a zvoleného měřicího rozsahu. Teprve poté je hodnota integrována. Algoritmus je znázorněn následujícím fragmentem kódu:

```
if (L3GD20_ReadRawData(NULL, NULL, &rawZ)) {
    angularRateX = ((float)rawX - offsetX) * L3DG20H_SENSITIVITY_LEVEL_500;
    angularRateY = ((float)rawY - offsetY) * L3DG20H_SENSITIVITY_LEVEL_500;
    angularRateZ = ((float)rawZ - offsetZ) * L3DG20H_SENSITIVITY_LEVEL_500;
    /*
     * Integration to get pitch, roll and yaw(heading) angles in degrees.
     * Value of angular rate is multiplied by 0.01 because period of this
     * task is 10 ms.
     */
    angleX += angularRateX * 0.01;
    angleY += angularRateY * 0.01;
    angleZ += angularRateZ * 0.01;
}
```

## 9.5 Měření okolní teploty

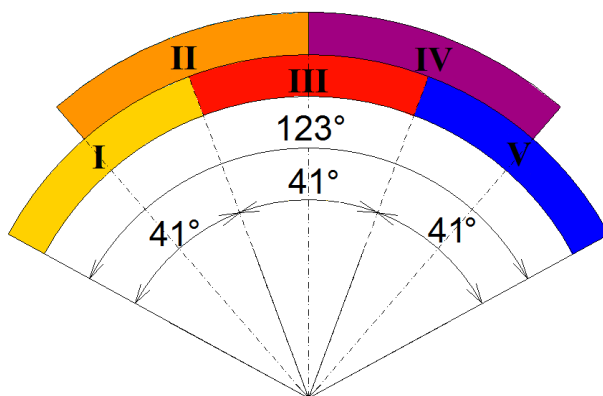
Okolní teplota je měřena v úkolu `TPA81_MeasureAmbientTemp()` s periodou 2 s. Hodnota okolní teploty je vyčítána po I2C sběrnici z příslušného registru senzoru TPA81 a následně uchována v globální proměnné. Odtud je poté dále využívána při detekci podezřelého objektu, kdy se vůči ní porovnává naměřená radiální teplota.

## 9.6 Měření povrchové teploty, detekce podezřelého objektu a překážky

Snímač povrchové teploty TPA81 i sonar HC-SR04 jsou umístěny na společné konzoli, kterou natáčí servo za účelem zvýšení zorného úhlu robota při zachování co nejjednodušší konstrukce a minimálního počtu senzorů. Při této konstrukci bylo třeba zajistit, aby v každé poloze serva proběhlo jak měření teploty tak i detekce překážky. Z tohoto důvodu jsou obě měření i polohování serva zapouzdřeny ve společném tásku `TPA81_MeasureRadiationTemp()`.

### 9.6.1 Krokovací režim serva

Servo ve výchozím stavu krokuje zleva doprava v 5 polohách, vždy po  $20,5^\circ$ . Velikost kroku vychází ze zorného úhlu senzoru TPA81, který činí v horizontální rovině  $41^\circ$ . Celkový zorný úhel je pak  $123^\circ$ . Velikost zorného úhlu je kompromisem s ohledem na rychlost robota a rozlišovací schopnost TPA81. Při větším zorném úhlu narůstá doba, za kterou se přesune servo s potřebným krokem z jedné krajní polohy do druhé. Přitom je třeba brát v úvahu prodlevu, která je v každém kroku nutná pro zapolohování serva, změření vzdálenosti sonarem HC-SR04 a vyčtení měřených hodnot přes I2C z TPA81. Jestliže nebylo s touto prodlevou uvažováno, velmi často se pak stávalo, že robot kolem hledaného objektu projel velmi těsně a nedetekoval ho, protože servo bylo v potřebný okamžik natočeno nesprávným směrem.



Obr. 46) Zorný úhel robota s vyznačením dílčích poloh serva

V počátcích vývoje bylo servo řízeno pouze ve třech polohách (levá krajní, prostřední a pravá krajní). Časem však bylo zjištěno, že pokud se hledaný objekt nacházel na rozhraní dvou poloh, senzor TPA81 ho nedokázal detekovat. Problém byl vyřešen přidáním dvou kroků serva, čímž došlo k částečnému překrytí dílčích zorných úhlů senzoru TPA81.

Algoritmus krokování obstarává fragment kódu níže. Poloha serva je reprezentována proměnnou `servoPosition`. Hodnota této proměnné odpovídá požadované střídě PWM signálu pro dosažení požadované polohy. PWM signál pro řízení serva je generován časovačem TPM2. Vzhledem k tomu, že krokování serva je prováděno v tásku `TPA81_MeasureRadiationTemp()`, nebylo žádoucí, aby byl tento tásk blokován aktivním čekáním v průběhu natáčení serva do nové polohy. Z toho důvodu se vytvoří časová prodleva před dalším spuštěním tohoto tásku a tásk se ukončí. Tím se zabraní zmíněnému blokování a v průběhu vytvořené prodlevy tak mohou být obslouženy další úlohy. Po uplynutí prodlevy se tento tásk spustí znovu, ale kód týkající se výpočtu nové polohy serva již nebude vykonán a

provede se místo něj (již při správném natočení serva) měření radiační teploty a měření vzdálenosti robota od překážky.

```
if (steppingMode) {
    if (servoLock == kServoUnlocked) {
        if (!servoOnPosition) {
            /* Move with servo from left to right */
            servoPosition += (TMP2_SERVO_POSITION_STEP);
            delay = 150U;
            if (servoPosition > TPM2_SERVO_MOST_RIGHT_POSITION) {
                servoPosition = TPM2_SERVO_MOST_LEFT_POSITION;
                delay = 350U;
            }
            /* Request on new servo position */
            TPA81_SetServoPosition(servoPosition);

            /* Make some delay for servo to move on required position */
            TASK_SetTaskCnt(4, delay);

            /* Update flag of new servo position */
            servoOnPosition = true;
            return;
        }
        servoOnPosition = false;
    }
}
```

### 9.6.2 Detekční algoritmus

Detekční algoritmus je úzce spjat s polohováním serva. Servo zpočátku pracuje v krokovacím režimu dle předchozí kapitoly a detekce podezřelého objektu se nachází ve stavu kHotSpotLost. V každé poloze serva se vyčte ze senzoru TPA81 všech 8 hodnot radiační teploty a tyto jsou uloženy v poli odpovídající velikosti. Toto pole je poté prohledáno a je nalezena pozice s nejvyšší teplotou.

```
/* Measure radiation temperature on the current servo position */
TPA81_MeasurePixelTemp();

/* Find pixel with highest temperature and remember its position */
for (uint8_t i = 0; i < 8; i++) {
    if (tPix[i] > tMaxAvg) {
        tMaxAvg = tPix[i];
        tMaxIdx = i;
    }
}
```



Následuje test nejvyšší zjištěné teploty oproti součtu okolní teploty a prahové teploty. Prahová teplota je v krokovacím režimu rovna 7 °C. Pokud je nejvyšší zjištěná radiační teplota větší než součet okolní a prahové teploty, senzor pravděpodobně zachytil podezřelý objekt. V takovém případě se dočasně deaktivuje krokovací režim serva, sníží se hodnota prahové teploty na 4 °C a sníží se dovolená vzdálenost robota od překážky na 5 cm.

```
if ((tMaxAvg >= (tAmbient + tThreshold))) {  
    /* Don't move with servo */  
    servoLock = kServoLocked;  
  
    /* Decrease temperature threshold, because of temperature fluctuation  
*/  
    tThreshold = 4;  
  
    /* Decrease allowed distance from obstacle */  
    SONAR_SetAllowedDistance(5);
```

Pokud v této konfiguraci potvrdí přítomnost podezřelého objektu další 3 po sobě jdoucí měření radiační teploty, tak se pravděpodobně jedná o skutečně podezřelý objekt. V takovém případě bude proveden výpočet korekce směru jízdy robota a stav detekce bude změněn z kHotSpotLost na kHotSpotDetected. V případě, že některé ze 3 měření přítomnost podezřelého objektu nepotvrdí, servo se vrátí do krokovacího režimu a robot bude pokračovat v dosavadním směru jízdy.

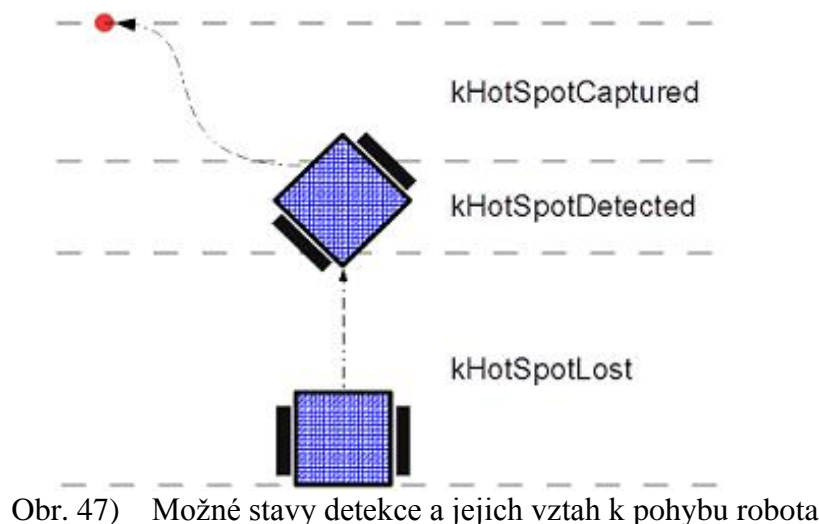
Bude-li v okamžiku detekce servo v prostřední pozici, provede se bez zastavení robota pouze jemná korekce směru dle pozice nejvyšší radiační teploty v poli všech 8 měřených hodnot, stav detekce bude změněn z kHotSpotDetected na kHotSpotCaptured, zruší se krokovací režim serva a nadále již bude pozice serva jemně regulována PI regulátorem.

Pokud bude v okamžiku detekce servo v jiné než prostřední pozici, korekce směru bude provedena i podle pozice serva, ve které k detekci došlo. Při této korekci se robot nejprve zastaví, servo se natočí do prostřední polohy, provede se natočení podvozku do správné polohy a robot se znovu rozjede. V ideálním případě bude ihned prvním měřením po natočení robota do nové polohy opět detekován podezřelý objekt. Robot tak bude okamžitě převeden do sledovacího režimu kHotSpotCaptured, kdy je pozice serva řízena regulátorem a ne krokovacím algoritmem. Může ale nastat situace, kdy po natočení robota žádný objekt detekován nebude. Toto je častý reálný případ, který nastává především v okamžiku, kdy je objekt detekován na delší vzdálenost. Po natočení robota totiž dojde k jinému rozložení teplot v jednotlivých měřících pixelech senzoru TPA81 a nejvyšší hodnota teploty již nemusí přesahovat stanovený práh. V takovém případě se tak opět aktivuje krokovací režim serva a robot se naslepo rozjede směrem, ve kterém je podezřelý objekt očekáván. Dříve či později pak robot opět detekuje hledaný objekt a mělo by již stačit provést pouze jemnou korekci směru.

Detekce skončí úspěšně, pokud robot nejpozději v okamžiku dosažení minimální vzdálenosti 5 cm od překážky v kterémkoliv pixelu TPA81 detekuje teplotu vyšší než 100 °C. V takovém případě byl úspěšně detekován předmět s nebezpečně vysokou povrchovou teplotou. Robot na znamení úspěšné detekce pošle přes sériovou linku informační zprávu,

rozsvítí se červená indikační LED dioda a ze bzučáku na Zumo Shield desce zazní zvukový signál.

Druhou možností jak může detekce skončit je ta, že robot sice zachytí podezřelý objekt, ale po přiblížení na nejkratší dovolenou vzdálenost povrchová teplota tohoto objektu nepřesahuje stanovený limit 100 °C. V takovém případě se objekt vyhodnotí jako obyčejná překážka, robot se otočí a bude pokračovat v hledání.



Obr. 47) Možné stavy detekce a jejich vztah k pohybu robota

### 9.6.3 Regulace polohy serva

Ve stavu `kHotSpotCaptured` přejde polohování serva z krokovacího režimu do režimu jemné regulace. Pro regulaci polohy serva byl implementován PI regulátor. Cílem regulace je, aby senzor TPA81 byl stále co nejpřesněji namířen na detekovaný objekt a dokázal ho sledovat jak při pohybu robota, tak i při pohybu samotného hledaného objektu.

Při hledání nejvyšší naměřené teploty v poli všech 8 změřených hodnot je uchována informace o pozici (index do pole v rozsahu 0 až 7), na které se nejvyšší teplota nachází. Pro účely regulace bylo definováno pomocné pole obsahující váhové ohodnocení jednotlivých pozic v poli hodnot radiační teploty. Podle pozice nejvyšší zjištěné radiační teploty se pak do regulátoru předá odpovídající váha. Žádanou veličinou je váha o hodnotě 0, která říká, že nejvyšší radiační teplota by měla ležet uprostřed pole všech změřených hodnot.

```
/* Array of servo heading angle correction values */
static int8_t tPosWiegght[8] = {-5, -3, -1, 0, 0, 1, 3, 5};

/* Servo position PI controller */
servoPosition = PID_ServoPositionController(tPosWiegght[tMaxIdx], 0);
TPA81_SetServoPosition(servoPosition);
```

Výstupem regulátoru je nová střída PWM signálu, který odpovídá požadované poloze serva. Zesilovací činitele proporcionální a integrační složky regulátoru byly nalezeny experimentálně. Bylo zjištěno, že derivační složka regulátoru na rychlosti regulačního procesu neměla příliš velký vliv a byla proto zanedbána. Rychlost regulace závisela především na maximální možné rychlosti použitého servo motoru.

```
uint16_t PID_ServoPositionController(int8_t actualPos, int8_t setpoint)
{
```

```
int32_t err, p, i, pwm;

/* Calculate actual error */
err = (int32_t) actualPos - (int32_t) setpoint;

/* P */
p = (int32_t) (ServoPid.pGain * (float) err);

/* I */
ServoPid.sumErr = ServoPid.sumErr + (float)err;
i = ServoPid.iGain * ServoPid.sumErr;

/* New output value of PI controller */
pwm = p + i;
pwm = TPM2_SERVO_CENTER_POSITION + pwm;

return ((uint16_t) pwm);
}
```

#### 9.6.4 Korekce směru jízdy v okamžiku detekce

Korekce směru jízdy je nutná proto, aby se robot natočil směrem k detekovanému objektu a mohl přijet blíže pro přesnější měření jeho povrchové teploty. Tato korekce je provedena pouze v okamžiku, kdy robot spatří podezřelý objekt (stav `kHotSpotDetected`). Výpočet sestává ze dvou kroků.

Prvním krokem je výpočet jemné korekce dle pozice pixelu, jímž byla detekována nejvyšší teplota. Výpočet vychází ze znalosti zorného úhlu jednoho pixelu v horizontální rovině, který činí  $5,12^\circ$ . V kódu je definováno pole o velikosti osmi prvků, které každému pixelu přiřazuje určité ohodnocení oproti ideálnímu stavu, kdy je podezřelý objekt detekován jedním z prostředních pixelů a korekční úhel je v takovém případě  $0^\circ$ . Vlastní výpočet jemné korekce bude proveden tak, že se podle indexu pixelu, který detekoval nejvyšší teplotu, vybere odpovídající ohodnocení a provede se jeho vynásobení s velikostí zorného úhlu jednoho pixelu.

```
/* Array of heading angle correction values */
static int8_t tCorr[8] = {-3, -2, -1, 0, 0, 1, 2, 3};

/* Calculate fine heading correction angle */
fineHeadingCorrection = (float)tCorr[tMaxIdx] * 5.12;
```

Druhým krokem je pak výpočet hrubé korekce dle polohy serva, v níž byl spatřen podezřelý objekt. Hrubá korekce se provádí pouze tehdy, pokud servo v okamžiku detekce není v centrální poloze. Při výpočtu se vychází ze znalosti pozorovacího úhlu robota a velikosti kroku serva viz obr.45. Při výpočtu se nejprve zjistí rozdíl skutečné polohy serva od centrální polohy (poloha serva je vyjádřena jako střída PWM). Znaménko výsledku značí, jestli je servo natočeno doprava či doleva a určuje tedy i požadovaný směr natočení robota. Poté se již provede převedení rozdílu na konkrétní hodnotu v úhlových stupních a k výsledku

se přičte hodnota jemné korekce. Výsledná korekce je předána do hlavního řídicího tásku, který spouští regulátor polohy. Současně je servo nastaveno do centrální polohy a je vytvořena časová prodleva před dalším spuštěním měřicího tásku nutná pro natočení podvozku do žádané polohy.

```
if (hotSpot == kHotSpotDetected) {
    /* Change heading only if hot spot isn't detected in servo center
       position */
    if (servoPosition != TPM2_SERVO_CENTER_POSITION) {
        delta = servoPosition - TPM2_SERVO_CENTER_POSITION;
        *pos = (41.0 / (float) TPM2_SERVO_POSITION_STEP) * delta;
        *pos += fineHeadingCorrection;

        /* Set servo into center position */
        TPA81_SetServoPosition(TPM2_SERVO_CENTER_POSITION);
        /* Create delay to make sure that Zumo is on the new position */
        TASK_SetTaskCnt(4, 750);
    } else {
        *pos = fineHeadingCorrection;
    }
}
```

### 9.6.5 Korekce směru jízdy ve sledovacím režimu

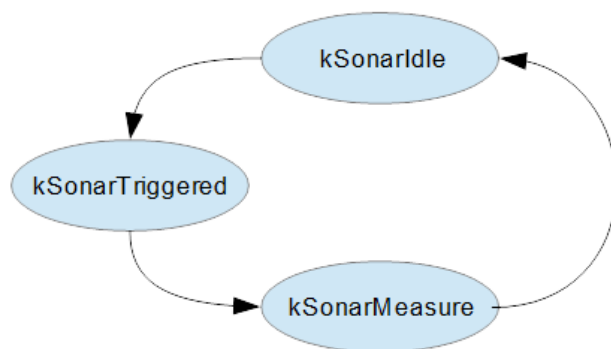
Ve sledovacím režimu (`kHotSpotCaptured`), kdy je robot již natočen směrem k podezřelému objektu a poloha serva je jemně řízena regulátorem, probíhá průběžná korekce, která zajistí změnu směru jízdy robota v případě pohybu detekovaného objektu. Opět se vychází z polohy serva. V případě, že servo není v centrální pozici, dojde ke snížení či zvýšení žádané hodnoty směru jízdy o  $0.75^\circ$ . Výsledná korekce je opět předána hlavnímu řídicímu tásku, který je vykonáván s periodou 10 ms. Periodickým prováděním této korekce pak dojde k plynulému zatáčení robota vlevo či vpravo podle směru pohybu detekovaného objektu.

Při tomto přístupu však nastal problém, kdy robot začal kličkovat v případě, že byla prováděna korekce směru při sebemenším vychýlení serva z centrální polohy. Tento problém byl vyřešen vytvořením pásma v okolí centrální polohy, ve kterém bude korekce vždy nulová.

```
/* If the hot spot is captured, calculate continuous Zumo heading angle
   correction according to servo position */
} else if (hotSpot == kHotSpotCaptured) {
    /* If servo position is greater than center position + deadband */
    if (servoPosition > (TPM2_SERVO_CENTER_POSITION + 400U)) {
        *pos = 0.75;
    }
    /* If servo position is greater than center position - deadband */
    } else if (servoPosition < (TPM2_SERVO_CENTER_POSITION - 400U)) {
        *pos = -0.75;
    }
    } else {
        *pos = 0;
    }
}
```

### 9.6.6 Měření vzdálenosti od překážky

Detekce překážky a měření vzdálenosti od ní probíhá podle scénáře uvedeného v kapitole 8.5.1 a celý proces je vyjádřen stavovým diagramem na následujícím obrázku:



Obr. 48) Stavový diagram procesu měření vzdálenosti od překážky

Měření je spuštěno voláním funkce `SONAR_MeasureDistance()`. Podmínkou měření je, že sonar je ve stavu `kSonarIdle`. Pokud v něm není, nebylo předchozí měření pravděpodobně správně zpracováno a v takovém případě se vyšle chybové hlášení přes sériovou linku a provede se reset sonaru. V případě, že tato podmínka splněna je, bude trigger pin nastaven do úrovně log. 1. Dále bude spuštěn kanál 1 časovače PIT pro odměření 10  $\mu$ s, což je minimální doba trvání pulzu na trigger pinu aby sonar spustil měření. Stav sonaru je poté změněn z `kSonarIdle` na `kSonarTriggered`, čímž se povolí měření délky echo impulzu kanálem 0 časovače TPM2. Po uplynutí 10  $\mu$ s bude vyvoláno přerušení časovače PIT a v jeho obsluze se nastaví úroveň na trigger pinu zpět do log. 0. Také se povolí přerušení časovače TPM2 při změně úrovně na echo pinu nebo přetečením čítače.

Časovač TPM2 je konfigurován jako volně běžící a je nastaven do input capture režimu. V okamžiku, kdy se objeví nástupná hrana na echo pinu, je vyvoláno přerušení od tohoto časovače a v obsluze přerušení se otestuje, zdali je sonar stále ve stavu `kSonarTriggered`. Pokud ano, bude vyčtena a v globální proměnné uchována aktuální hodnota čítače a stav sonaru je převeden do stavu `kSonarMeasure`. Pokud se nyní objeví sestupná hrana na echo pinu, bude opět vyčten aktuální stav čítače a z rozdílu této a předchozí hodnoty se určí, kolik cyklů čítač načítal mezi nástupnou a sestupnou hranou. Stav sonaru je po dokončení měření změněn na `kSonarIdle`, přerušení od TPM2 je zakázáno a vše je tak připraveno na další měření.

Před vlastním výpočtem vzdálenosti od překážky (v jednotkách délky) je třeba převést naměřenou délku echo pulzu, která je vyjádřena v počtu cyklů TPM2, na čas. Podle nastavení časovače trvá jeden cyklus 0,333  $\mu$ s. Vynásobením počtu cyklů délkou trvání jednoho cyklu je zjištěna doba trvání echo impulzu a vzdálenost od překážky je pak stanovena dle vztahu uvedeného v kapitole 6.5.1. Vypočtená hodnota vzdálenosti [cm] je uchována pro další zpracování v globální proměnné.

Při výpočtu vzdálenosti jsou zanedbány faktory jako teplota či hustota vzduchu, ve kterém se šíří ultrazvukové vlnění emitované sonarem. Předpokladem je, že robot je využíván v prostředí s normální pokojovou teplotou a nejsou kladeny žádné přísné požadavky na přesné určení vzdálenosti. Z toho důvodu je rychlost zvuku, která je zahrnuta ve výpočtu, definována jako konstanta 343  $\text{m}\cdot\text{s}^{-1}$ . Při testování sonaru bylo zjištěno, že při tomto nastavení je

vzdálenost 50 cm změřena s přesností 5 mm. Tato přesnost je pro navrhovaný robot postačující.

## 9.7 Komunikace

Komunikace robota s okolím probíhá po sériové lince. K přenosu je využita periferie UART0. Výhodou použitého vývojového kitu FRDM-KL25Z je, že obsahuje debug rozhraní OpenSDA. Toto rozhraní slouží pro ladění aplikací na MCU přes USB kabel bez externího JTAG debuggeru. Současně toto rozhraní emuluje sériový COM port, na který je možné se připojit z host PC. Na emulovaný port jsou vyvedeny Rx/Tx signály rozhraní UART0 a současně jsou ty stejné signály vyvedeny na piny PTA1 a PTA2. K těmto pinům je připojen Bluetooth modul a je tak možné komunikovat jak přes USB kabel (emulovaný sériový port), tak i přes Bluetooth.

Veškeré API pro ovládání periferie UART0 je implementováno v modulu `uart.c`. Komunikace probíhá v interrupt režimu. Rychlost sériové komunikace byla nastavena na 115200 bps. Využívá se jeden start bit, parita nikoliv.

Data pro odeslání a data přijatá jsou vkládána do oddělených kruhových bufferů, odkud jsou dále zpracovávána. Pro formátovaný výstup dat byla využita knihovna `xformatc` [32]. Tato knihovna implementuje funkci `xvformat()`, což je odlehčená verze standardní funkce `printf()`. Funkce `xvformat()` převede všechny vstupní parametry na řetězec a ten je následně prostřednictvím funkce `UART0_SetTxBuffer()` vložen do odesílacího bufferu a odtud následně odvyšlán.

Komunikační rozhraní je využíváno především pro asynchronní vysílání provozních dat a informací o stavu detekce. Robot akceptuje dva jednoduché příkazy pro povolení rozjezdu a zastavení. V implementovaném jednoduchém protokolu je využit znak `<` jako počáteční znak a oddělovač každého přijatého příkazu.

Tab 7) Popis robotem akceptovaných příkazů

Příkaz	Význam
<code>&lt;start</code>	Povolení rozjezdu
<code>&lt;stop</code>	Zastavení robota

Bohužel komunikační část práce se nepodařilo zcela dokončit. Plánována byla implementace protokolu, který by umožňoval manuální řízení robota a zasílání konfiguračních dat z mobilní aplikace. Z časových důvodů však na realizaci nedošlo.



## 10 ZHODNOCENÍ A DISKUZE

Robot byl testován s využitím podezřelého objektu, který tvořila automobilová žárovka H5, na níž byla nasunuta hliníková trubka s vnějším průměrem 20 mm. Žárovka byla napájena 5 V zdrojem. Optickým pyrometrem bylo změřeno, že povrchová teplota hliníkové trubky je při pokojové teplotě a tímto napájecím napětí přibližně 120 °C. Robot je schopen tuto žárovku detekovat přibližně na vzdálenost 1 m. Experimentálně byla testována detekce elektrické varné plotýnky o průměru 280 mm rozehřáté na 250 °C. Tuto dokázal robot odhalit na vzdálenost přibližně 4 m.

Problémem použitého infračerveného termočlánku TPA81, je omezené pásmo ve vertikální rovině (5,12°) ve kterém, je schopen senzor měřit. Hledaný objekt proto musí být umístěn ve správné výšce, jinak ho senzor nedetekuje. Také řízení serva po sběrnici I2C prostřednictvím TPA81 se ukázalo jako nevhodné z důvodu nedostatečného rozlišení jednotlivých poloh. Jako daleko vhodnější se ukázalo použít PWM výstup mikrokontroléru a střidu PWM řídit regulátorem. Teplota naměřená v jednotlivých měřících pixelech senzoru mírně kolísá i v případě, že senzor zachycuje předmět s konstantní teplotou. Tato vlastnost není při zvolené robotické úloze kritická. V případě, že by bylo nutné měřit teplotu přesně, musel by být použit kvalitnější senzor.

Při návrhu dalšího robota by bylo vhodnější použít více senzorů pro detekci překážky. Navržený robot je vybaven pouze jedním ultrazvukovým sonarem umístěným na otočné konzoli. Ukázalo se však, že tuto konstrukci limituje rychlost serva, na kterém je konzola přimontována. V každé poloze serva je totiž nutné vytvořit určitou prodlevu pro úspěšné vyslání a přijetí odražené ultrazvukové vlny a tento proces je nutné opakovat v každé poloze, aby byl pokryt co největší prostor před a po stranách robota. Pro řešenou robotickou úlohu je však důležité, aby robot za co nejkratší ujetou vzdálenost prohledal co největší prostor a tedy změřil co nejvíce vzorků radiační teploty v co největším počtu poloh serva. Tím se minimalizuje možnost, že robot kolem hledaného objektu projede, aniž by ho detekoval. Vhodnější by bylo použít samostatné senzory pro nezávislou detekci na levé a pravé straně robota a jeden senzor pro detekci překážky před robotem ve směru jízdy. Detekce překážek by pak mohla probíhat s delší periodou, aniž by byla ovlivněna rychlost polohování serva. Ta by pak byla pouze omezena rychlostí samotného serva. Měření radiační teploty a její vyčtení po sběrnici I2C není v tomto případě limitující, protože tato operace je mnohonásobně rychlejší než vlastní přesun serva z jedné polohy do další.

Problematickou částí bylo řízení otáčení robota na místě a ladění směrového regulátoru, kdy se projevila nevýhoda pásového podvozku. Pro sebemenší otočení pásového podvozku je nutné do obou motorů dodat velký příkon, aby došlo k odtržení ze statické pozice. Implementovaný PID regulátor potom vykazoval buď příliš velký překmit, nebo naopak nedokázal doregulovat natočení podvozku do žádané pozice v uspokojivém čase. Z tohoto důvodu bylo učiněno opatření, kdy minimální úhel, o který se robot může otočit, byl stanoven na 30° a sumace regulační odchylky pro integrační složku regulátoru je na počátku regulačního procesu deaktivovaná a aktivuje se až v okamžiku, kdy je regulační odchylka v intervalu  $< -5^\circ; 5^\circ >$ .





## 11 ZÁVĚR

Cílem této diplomové práce bylo navrhnout, zkonstruovat a naprogramovat mobilního autonomního robota, který bude schopen ve svém okolí vyhledávat objekty s potenciálně nebezpečně vysokou povrchovou teplotou. Tento cíl byl splněn a výsledkem je funkční model robota. Práce má svoji teoretickou a praktickou část.

V teoretické části, která je rešeršního charakteru, jsou popsány základní typy robotických podvozků, které bylo možné použít s ohledem na zadání a robotickou úlohu. Jedná se o podvozky kolové a pásové. Pro robota je klíčovou vlastností bezkontaktní skenování teploty předmětů v okolí. Proto následuje kapitola věnovaná teorii bezkontaktního měření teploty, ve které jsou popsány základní fyzikální principy a typy snímačů používaných pro tuto metodu měření. Další kapitoly pak popisují jednotlivé subsystemy robota a prvky, ze kterých bylo možné vycházet při návrhu jednotlivých částí robota včetně jejich principů činnosti.

Praktická část práce se zabývá návrhem robota a popisem konkrétních komponent, které byly vybrány pro jeho realizaci. Základem je pásový podvozek Zumo Robot. Jako pohon slouží dva stejnosměrné motory s převodovkou. Pro měření otáček těchto motorů a za účelem jejich řízení byly použity dva inkrementální enkodéry pracující na magnetickém principu. Směr jízdy robota je měřen MEMS gyroskopem. Podvozek je dále osazen servo motorem. Tento motor nese konzoli, na níž je připevněn ultrazvukový sonar HC-SR04 pro detekci překážky a infračervený termočlánek TPA81 pro bezkontaktní snímání radiační teploty. Natáčením této konzole je tak zvýšen zorný úhel robota. Pro řízení robota byl zvolen vývojový kit FRDM-KL25Z firmy Freescale.

Nejdůležitější částí práce byla implementace řídicího algoritmu a detekce podezřelých objektů. Pro nezávislé řízení rychlosti každého z obou motorů byly implementovány dva PID regulátory, kterým zpětnou vazbu poskytují inkrementální magnetické enkodéry. K těmto regulátorům byl dále implementován nadřazený regulátor pro řízení směru jízdy na základě dat získaných z gyroskopu. Žádaná hodnota směru jízdy je výstupem algoritmu pro detekci podezřelých objektů. Součástí tohoto algoritmu je PI regulátor, který řídí polohu servomotoru tak, aby infračervený termočlánek TPA81 připevněný na motor byl vždy v optimální poloze vzhledem k hledanému objektu. Dle aktuální polohy servomotoru a teplotního scanu je průběžně vyhodnocována korekce směru jízdy. Robot tak dokáže lokalizovat nehybné i pronásledovat pohybující se předměty, které oproti okolnímu prostředí vykazují podezřele zvýšenou povrchovou teplotu.

V plánu také byla implementace mobilní aplikace pro systém Android, pomocí které by bylo možné přijímat bezdrátově diagnostické zprávy či převést robot do režimu manuálního řízení a robot vzdáleně ovládat přes Bluetooth. Z časových důvodů však tato aplikace nebyla dokončena a zůstala pouze ve stavu rozpracovaného prototypu. Komunikace je tak možná přes USB kabel, který emuluje virtuální sériový port. Druhou možností je některá ze standartních mobilních aplikací, které poskytují terminálové okno pro zasílání zpráv přes rozhraní Bluetooth.

Kombinace podvozku Zumo Robot a vývojového kitu FRDM-KL25Z se ukázala jako velice zajímavá a do budoucna poskytuje velký prostor k řešení dalších robotických úloh. V případě nasazení této platformy do výuky by bylo možné použít ji pro úlohy jako je

například řešení bludiště, sledování čáry, nebo zápasy Mini Sumo. Robot by však mohl být nasazen i do sofistikovanějších úloh, které vyžadují vzájemnou spolupráci několika robotů. Zajímavým rozšířením může být i montáž kamery pro zpracování obrazu nebo dokovací stanice pro automatické dobíjení akumulátorů. S ohledem na budoucí použití byl řídicí firmware navržen modulárně tak, aby každý modul zapouzdřoval určitou periférii MCU nebo externí součástku. Možným rozšířením těchto modulů může být například podpora DMA, pokud ji daná periférie MCU podporuje, což by vedlo k úspoře části procesorového času. Dále by bylo možné implementovat komplexní knihovnu pro ovládání veškerých součástek dostupných na podvozkové desce. Z pohledu HW by možné rozšíření mohlo spočívat v návrhu a výrobě nové podvozkové desky, na které by byl přímo osazen mikrokontrolér a senzory. Navíc by poskytovala konektory pro připojení dalších externích součástek bez kabelového propojení.

## 12 SEZNAM POUŽITÝCH ZDROJŮ

- [1] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. 1. vyd. Praha: BEN technická literatura, 2005, 243 s. ISBN 80-7300-141-1.
- [2] VOŽENÍLEK, Ladislav. *Kurs elektrotechniky*. 1. vyd. Praha: SNTL, 1976, 386 s. Řada elektrotechnické literatury.
- [3] HALLIDAY, D., RESNICK, R., WALKER, J.: *Fyzika*. 1. vyd. Brno: VUTIUM, 2000
- [4] *Bezdotykové měření teploty*. Soubor technických informací poskytovaných firmou OMEGA: Zpravodaj pro měření a regulaci. Vydání č. 1, 2. Dostupné z: [http://www.omegaeng.cz/literature/PDF/techinfo\\_1.pdf](http://www.omegaeng.cz/literature/PDF/techinfo_1.pdf)
- [5] GRUNER, Klaus. *Principles of Non-Contact Temperature Measurement*. RAYTEK [online]. Rev. B 2003 [cit. 2016-01-04]. Dostupné z: [http://support.fluke.com/raytek-sales/Download/Asset/IR\\_THEORY\\_55514\\_ENG\\_REVB\\_LR.PDF](http://support.fluke.com/raytek-sales/Download/Asset/IR_THEORY_55514_ENG_REVB_LR.PDF)
- [6] *Tepelné záření. Záření absolutně černého tělesa*. Učební text k předmětu UFY102. Univerzita Karlova v Praze, Matematicko-fyzikální fakulta. 2006 [cit. 2016-01-04]. Dostupné z: <http://alma.karlov.mff.cuni.cz/ufy102/Planckuv%20zakon.pdf>
- [7] *75:1 Micro Metal Gearmotor HP with Extended Motor Shaft*. Pololu: Robotics & Electronics. [online]. [cit. 2015-10-13]. Dostupné z: <https://www.pololu.com/product/2215>
- [8] ČEJKA, Š. *FITKIT jako řídicí jednotka krokového motoru*. Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, 2011 [cit. 2016-01-20]. 48 s. Vedoucí bakalářské práce Ing. Josef Strnadel, Ph.D.
- [9] [9] ŘEZÁČ, Kamil. *Krokové motory*. Robotika.cz. [online]. 28.10.2002 [cit. 2016-01-20]. Dostupné z: <http://robotika.cz/articles/steppers/cs>
- [10] *Magnetic Encoder Pair Kit for Micro Metal Gearmotors*. Pololu: Robotics & Electronics. [online]. [cit. 2015-10-13]. Dostupné z: <https://www.pololu.com/product/2598>
- [11] *Zumo Robot for Arduino, v1.2*. Pololu: Robotics & Electronics. [online]. [cit. 2016-01-05]. Dostupné z: <https://www.pololu.com/product/2510>
- [12] *Schematic diagram of Zumo Robot for Arduino, v1.2*. Pololu: Robotics & Electronics. [online]. [cit. 2016-01-05]. Dostupné z: [https://www.pololu.com/file/0J591/zumo\\_shield\\_schematic.pdf](https://www.pololu.com/file/0J591/zumo_shield_schematic.pdf)
- [13] *FRDM-KL25Z*: Freescale Freedom Development Platform for Kinetis KL14, KL15, KL24, KL25 MCUs. Freescale [online]. 2015 [cit. 2015-10-14]. Dostupné z:

<http://www.freescale.com/tools/embedded-software-and-tools/hardware-development-tools/freedom-development-boards/freescale-freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z>

- [14] *KL25P80M48SF0RM*: KL25 Sub-Family Reference Manual. Freescale [online]. Rev. 3 2012 [cit. 2015-10-15]. Dostupné z: [http://cache.freescale.com/files/32bit/doc/ref\\_manual/KL25P80M48SF0RM.pdf](http://cache.freescale.com/files/32bit/doc/ref_manual/KL25P80M48SF0RM.pdf)
- [15] HRBÁČEK, J. *Návrh a realizace regulátoru otáček stejnosměrného motoru pro mobilní robot*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008 [cit. 2016-01-05]. 49 s. Vedoucí bakalářské práce Ing. Jiří Krejsa, Ph.D.
- [16] LOCKER, Martin. *Inkrementální enkodér*. Robot klub Rychnov. [online]. 17.7.2009 [cit. 2016-01-16]. Dostupné z: <http://robotika.vosrk.cz/guide/sensors/decode/cs>
- [17] *Speed of sound*. Wikipedia: the free encyclopedia. [online]. [cit. 2016-01-03]. Dostupné z: [https://en.wikipedia.org/wiki/Speed\\_of\\_sound](https://en.wikipedia.org/wiki/Speed_of_sound)
- [18] *Working or Operating Principle of DC Motor*. Online Electrical Engineering. [online]. [cit. 2016-01-03]. Dostupné z: <http://www.electrical4u.com/working-or-operating-principle-of-dc-motor/>
- [19] *TPA81 Thermopile Array*. Robot Electronics: Technical Specification. [online]. [cit. 2016-01-04]. Dostupné z: <http://www.robot-electronics.co.uk/htm/tpa81tech.htm>
- [20] VOJÁČEK, Antonín. *Integrované MEMS GYROSKOPY*. HW server: Automatizace. [online]. 11.10.2009 [cit. 2016-01-11]. Dostupné z: <http://automatizace.hw.cz/integrované-mems-gyroskopy>
- [21] MALÝ, Martin. *Magnetický rotační enkodér AS5040*. vyvoj.hw.cz: Profesionální elektronika. [online]. 21.5.2005 [cit. 2015-12-12]. Dostupné z: <http://vyvoj.hw.cz/soucastky/magneticky-rotacni-encoder-as5040.html>
- [22] *Encoder Technology – Details & Descriptions*. Leine Linde. [online]. [cit. 2015-12-12]. Dostupné z: [http://www.leinelinde.com/PageFiles/2266/1073559-01\\_Encoder\\_technology\\_eng.pdf](http://www.leinelinde.com/PageFiles/2266/1073559-01_Encoder_technology_eng.pdf)
- [23] PASOLINI, Fabio. *MEMS Accelerometers, Gyroscopes, and Geomagnetic Sensors - Propelling Disruptive Consumer Applications*. Digi-Key: Electronics. [online]. 11.4.2011 [cit. 2016-01-11]. Dostupné z: <http://www.digikey.com/en/articles/techzone/2011/apr/mems-accelerometers-gyroscopes-and-geomagnetic-sensors---propelling-disruptive-consumer-applications>
- [24] *L3GD20H MEMS motion sensor*: Three-axis digital output gyroscope. Datasheet. STMicroelectronics [online]. Rev. 2 2013 [cit. 2016-1-14]. Dostupné z: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00060659.pdf>

- [25] *Tilt measurement using a low-g 3-axis accelerometer*. Application note: AN3182. STMicroelectronics [online]. Rev. 1 2010 [cit. 2016-1-26]. Dostupné z: [http://www.st.com/web/en/resource/technical/document/application\\_note/CD00268887.pdf](http://www.st.com/web/en/resource/technical/document/application_note/CD00268887.pdf)
- [26] VOLF, Jaromír. *Taktilní senzory pro automatizaci*. Automa, 2008, č. 7, s. 16-19. ISSN: 1210-9592. [cit. 2016-01-04]. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/37544.pdf>
- [27] *Omni-Directional Wheels*. OMNIX Technology: Revolution in motion. [online]. [cit. 2016-02-02]. Dostupné z: [http://www.omnixtechnology.com/direct\\_components.html](http://www.omnixtechnology.com/direct_components.html)
- [28] *TALON*. Army-Technology.com: The top five military robots. [online]. [cit. 2016-02-02]. Dostupné z: <http://www.army-technology.com/features/featurearmed-aware-and-dangerous-the-top-five-military-robots/featurearmed-aware-and-dangerous-the-top-five-military-robots-2.html>
- [29] MIKÉSKA, Zdeněk. *Specifikace rádiové části systému Bluetooth*. Elektrorevue: Časopis pro elektrotechniku. [online]. 14.1.2004 [cit. 2016-04-08]. Dostupné z: <http://www.elektrorevue.cz/clanky/04003/index.html>
- [30] VOJÁČEK, Antonín. *ZigBee – novinka na poli bezdrátové komunikace*. vyvoj.hw.cz: Profesionální elektronika. [online]. 8.6.2005 [cit. 2016-04-08]. Dostupné z: <http://vyvoj.hw.cz/navrh-obvodu/rozhrani/zigbee-novinka-na-poli-bezdratove-komunikace.html>
- [31] PINDER, Christian. *Game Programming: Random Number Generation*. Personal website of Christian Pinder. [online]. [cit. 2016-04-08]. Dostupné z: <https://www.christianpinder.com/articles/pseudo-random-number-generation/>
- [32] VIARA, Mario. *Library xformatc (printf for embedded system with floating point support)*. Dostupné z: <https://github.com/MarioViara/xprintfc>
- [33] VDOLEČEK, F. *Technická měření*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2002 [cit. 2016-01-05]. Dostupné z: <http://autnt.fme.vutbr.cz/lab/a1-731a/ETC.pdf>
- [34] ZUTH, D.; MATOUŠEK, R.; MARADA, T. Expression Theoretical Temperature of Objects. In MENDEL 2009. Mendel Journal series. 1. Brno: BUT, 2009. s. 298-302. ISBN: 978-80-214-3884- 2. ISSN: 1803- 3814.
- [35] *Serial Port Bluetooth Module (Master/Slave) : HC-05*. [online]. 2015 [cit. 2016-5-7]. Dostupné z: [http://wiki.iteadstudio.com/Serial\\_Port\\_Bluetooth\\_Module\\_\(Master/Slave\)\\_:HC-05](http://wiki.iteadstudio.com/Serial_Port_Bluetooth_Module_(Master/Slave)_:HC-05)



## 13 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

### 13.1 Seznam symbolů a zkratk

A/D	Analog to Digital Converter
ALU	Aritmetical and Logical Unit
ARM	Advanced RISC Machine
CAN	Controller Area Network
CPU	Central processor unit
D/A	Digital to Analog Converter
DPS	Degrees Per Second
I2C	Inter-Integrated Circuit
LED	Light Emitting Diode
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical Systems
PI	Proportional-Integral Controller
PID	Proportional-Integral-Derivative Controller
PWM	Pulse Width Modulation
RTC	Real Time Clock
SPI	Serial Peripheral Interface
TPM	Timer/PWM Module
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver and Transmitter

**13.2 Seznam obrázků**

Obr. 1)	Balancují robot.....	19
Obr. 2)	Detail všesměrového kola.....	20
Obr. 3)	Vojenský robot TALON s pásovým podvozkem .....	20
Obr. 4)	Infračervená oblast elektromagnetického záření .....	21
Obr. 5)	Planckův vyzařovací zákon .....	23
Obr. 6)	Infračervený termočlánek firmy Omega.....	26
Obr. 7)	Měřený objekt musí zcela vyplňovat měřenou plochu .....	26
Obr. 8)	Elementární DC motor.....	29
Obr. 9)	Průběh PWM signálu se střídou 30% a průběhy napětí a proudu na motoru .....	30
Obr. 10)	Pracovní režimy stejnosměrného motoru .....	31
Obr. 11)	Jednokvadrantový budič .....	31
Obr. 12)	Dvoukvadrantový budič.....	31
Obr. 13)	Čtyřkvadrantový budič .....	32
Obr. 14)	Řez hybridním KM .....	33
Obr. 15)	Unipolární řízení .....	34
Obr. 16)	Bipolární řízení .....	34
Obr. 17)	Jednofázové řízení .....	35
Obr. 18)	Dvoufázové řízení.....	35
Obr. 19)	Princip řízení s polovičním krokem.....	35
Obr. 20)	Modelářské servo.....	36
Obr. 21)	Magnetický enkodér v integrovaném provedení .....	38
Obr. 22)	Absolutní enkodér.....	38
Obr. 23)	Kvadrurní signál generovaný inkrementálním enkodérem.....	39
Obr. 24)	Struktura MEMS gyroskopu.....	40
Obr. 25)	Měření náklonu .....	41
Obr. 26)	Princip optického taktilního senzoru s mikroohybem .....	42
Obr. 27)	Podvozek Zumo Robot v1.2 .....	48
Obr. 28)	Pohled na Zumo Shield a šasi s DC motory .....	49
Obr. 29)	Použitý DC motor a enkodér Pololu připevněný k motoru .....	50
Obr. 30)	Vývojová deska FRDM-KL25Z .....	52
Obr. 31)	Ultrazvukový sonar HC-SR04 .....	52
Obr. 32)	Časová posloupnost signálů pro práci se sonarem .....	53
Obr. 33)	Převodník napěťových úrovní pro připojení sonaru k MCU .....	53
Obr. 34)	Souřadný systém robota.....	54
Obr. 35)	Thermopile Array TPA81 .....	55
Obr. 36)	Zorné pole snímače.....	55
Obr. 37)	Změna snímané plochy v závislosti na vzdálenosti .....	56
Obr. 38)	Bluetooth modul HC-05 .....	57
Obr. 39)	Sendvičová konstrukce robota. Nahoře řídicí deska FRDM-KL25Z, uprostřed rozšiřující deska a dole podvozková deska Zumo Shield.....	58
Obr. 40)	Fotografie navrženého robota a testovacího přípravku pro simulaci předmětu s vysokou povrchovou teplotou (žárovka H5).....	58
Obr. 41)	Stavový diagram robota .....	61
Obr. 42)	Blokové schéma regulace jízdy robota .....	63
Obr. 43)	Blokové schéma regulace otáčení robota .....	66



Obr. 44)	Průběh regulace při skokové změně žádané polohy (natočení) .....	66
Obr. 45)	Vliv kalibrace na gyroskop a potlačení driftu.....	70
Obr. 46)	Zorný úhel robota s vyznačením dílčích poloh serva .....	71
Obr. 47)	Možné stavy detekce a jejich vztah k pohybu robota .....	74
Obr. 48)	Stavový diagram procesu měření vzdálenosti od překážky .....	77

**13.3 Seznam tabulek**

Tab 1)	Mechanické parametry podvozku Zumo.....	48
Tab 2)	Parametry použitých DC motorů při 6 V .....	50
Tab 3)	Parametry použitých enkodérů.....	51
Tab 4)	Popis interních registrů senzoru TPA81 .....	56
Tab 5)	Parametry Bluetooth modulu HC-05 .....	57
Tab 6)	Stručný význam definovaných tasků .....	60
Tab 7)	Popis robotem akceptovaných příkazů.....	78

## 14 SEZNAM PŘÍLOH

### PŘÍLOHA 1:CD

- 1.1 Text práce ve formátu PDF
- 1.2 Projekt s řídícím firmware pro vývojové prostředí Kinetis Design Studio 3.0.0
- 1.3 Videoklip jízdy robota



## **PŘÍLOHY**